11-28-2014 12:00 AM

Multivariate Spatial Visualization using GeoIcons and Image Charts

Bo Shan
The University of Western Ontario

Supervisor
Micha Pazner
The University of Western Ontario

Graduate Program in Geography
A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

MULTIVARIATE SPATIAL VISUALIZATION USING GEOICONS AND IMAGE CHARTS

(Thesis format: Monograph)

by

Bo Shan

Graduate Program
in
Geography

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

Spatial databases are growing in size and complexity, yet current visual data mining methods are challenged when it comes to multivariate spatial data. The specific research question addressed in this thesis is: how can spatial multivariate data be effectively visualized using an icon based non-fused co-visualization approach? The thesis presents a Python based design and implementation of a visualization program termed *GeoIcon Viewer*. The program incorporates two different visualization methods: GeoIcon Image Map and Region-of-Interest Image Layers Chart. The GeoIcon Image Map technique uses an icon to co-visualize up to nine attributes at a single location. The Region-of-Interest Image Layers Chart method uses a small multiples approach to support the GeoIcon Image Map technique for data with negligible value differences. The thesis demonstrates the successful implementation of the GeoIcon Viewer with a case study involving remote sensing digital image analysis of a copper deposit. With the two visualization methods and eight input attributes, the GeoIcon Viewer generated real time interactive visualization outputs that can aid a user in multivariate spatial data mining.

Keywords: Visualization, Multivariate, Spatial Data, Icons, Geographic Information Systems, GIS, Remote Sensing, GeoIcon

Contents

Certificate of Examination **ii**

Abstract **ii**

Table of Contents **iii**

List of Figures **vi**

List of Tables **ix**

1 Introduction **1**

1.1 Problem 1

1.2 Objective 3

1.3 Thesis Organization 4

2 Literature and Background **6**

2.1 Knowledge Discovery from Databases 6

2.1.1 Input Data: Non-Spatial and Spatial Data 7

2.2 Data Mining Stage of KDD 10

2.2.1 The Computational Approach 10

2.2.2 The Visualization Approach 11

2.2.3 Exploratory Visual Analysis 13

2.3 Multivariate Data Visualization 17

2.4 Summary 24

3 System Design **25**

3.1 Introduction 25

3.2 Overall System Design 26

3.2.1 Input Data Requirement 27

3.2.2 Visualization Methods 28

3.2.3 Interaction Techniques 28

3.2.4 Resulting GUI Design 30

3.3 Components Design and Specifications 31

3.3.1 Icon Design 35

3.3.2 Region-of-Interest Image Layers Chart 38

3.3.3 Summary and the Next Step 39

4 Software Implementation **41**

4.1 Implementation Process 41

4.2 Implementation: Program Paradigm 42

4.3 Data Flow 43

4.4 Component Development and Implementation 46

4.4.1 Python Libraries 46

4.4.2 Overview Window: The Input Data 47

4.4.3 Visualization Processors 54

4.4.4 Display of Visualization Outputs 61

4.5 Components Integration 63

5 Demonstration and Evaluation **65**

5.1 Case Study and Data 66

5.1.1 Alunite 77

5.1.2 Calcite 77

5.1.3 Kaolinite and Hydroxide 77
5.1.4 Quartz 78
5.1.5 Carbonate, Mafic, and Vegetation 78
5.1.6 Input Data for GeoIcon Viewer 79
5.2 Results and Analysis 79
5.2.1 Overview Window 80
5.2.2 Colour Chooser and Control/Legend Window 81
5.2.3 Visualization Window: Icon Design 81
5.2.4 Visualization Outputs: Three Attributes 83
5.2.5 Visualization Outputs: Five Attributes 89
5.2.6 Visualization Outputs: Eight Attributes 94
5.3 Program Evaluation 98

6 Summary and Future Work **100**

6.1 Future Improvements 103

Curriculum Vitae **110**

List of Figures

1.1 The GeoIcon Viewer program 4

2.1 Statistical profiles of four datasets with different patterns 12

2.2 Dataset plots 12

2.3 Hertzsprung Russell Diagram 13

2.4 Visualization technique classification chart 16

2.5 Bristle map approach for vector data 19

2.6 IconMapper icon design 20

2.7 IconMapper output visual clutter 22

2.8 Pixel bar chart design 23

2.9 Pixel bar chart for consumer spendings 23

3.1 Waterfall development model 25

3.2 Feature requirements translated into program components 26

3.3 Graphic User Interface conceptual design 31

3.4 Grayscale low contrast image 32

3.5 Overview image comparison 33

3.6 GeoIcon Viewer icon design 35

3.7 Icon element colour customization 38

3.8 Region-of-Interest Image Layers Chart method design 39

4.1 Implementation stages of the Waterfall model 42

4.2 GeoIcon Viewer system algorithm 44
4.3 Overview Window and associated components 47
4.4 Open Image: creating false colour composite 48
4.5 Overview Window class diagram 51
4.6 Visualization Processors and associated components 54
4.7 Control/Legend Window class diagram 55
4.8 GeoIcon class diagram 56
4.9 Icon element construction algorithm 57
4.10 Region-of-Interest Image Layers Chart class diagram 59
4.11 Visualization Window and associated processes 61
4.12 Visualization Window class diagram 62

5.1 An image map of the case study area 67
5.2 Alunite distribution map 69
5.3 Calcite distribution map 70
5.4 Kaolinite distribution map 71
5.5 Hydroxide distribution map 72
5.6 Quartz distribution map 73
5.7 Carbonate distribution map 74
5.8 Mafic distribution map 75
5.9 Vegetation distribution map 76
5.10 Overview Window component evaluation 80
5.11 Control/Legend Window implementation result 81
5.12 GeoIcon Viewer icon design implementation result 82
5.13 Icon image map for three attributes with input data range of 2 84
5.14 Icon image map for three attributes with input data range of 15 . . . 87
5.15 Region-of-Interest Image Layers Chart method with three attributes . 88
5.16 Icon image map for five attributes with input data range of 2 90

5.17 Icon image map for five attributes with input data range of 15 91
5.18 Region-of-Interest Image Layers Chart with five attributes 93
5.19 Icon image map for eight attributes with a non threshold image . . . 96
5.20 Region-of-Interest Image Layers Chart with eight attributes 97

List of Tables

5.1 Mineral index equations 66

Chapter 1

Introduction

1.1 Problem

Advances in information technology such as data collection, distribution, and storage have brought many improvements to geographic information systems. Modern satellite sensors with faster revisit time collect vast amount of high resolution data. Along with advancements in personal mobile sensors, the amount of spatial data is growing rapidly. Not only are the sizes of databases increasing, but the types of data measured and collected are more numerous. The result is the emergence of spatial databases of unprecedented size and complexity.

The field of data mining and Knowledge Discovery from Database (KDD) was developed to extract patterns and knowledge from vast amounts of data. Spatial data contain spatial information that correlates to locations on Earth or other physical spaces, and require special data mining techniques other than that of alpha-numeric data due to concepts such as spatial autocorrelation, spatial heterogeneity, and direction.

Spatial databases are of little value without proper methods to extract the information contained in them. A specialized branch of KDD, Geographic Knowledge Discovery from Database (GKDD) which uses geocomputational and geovisualization methods, was developed in response to the growing complexity of spatial databases (Miller and Han, 2009).

While data mining via (geo)computational methods are efficient and comprehensive, current methods are not flexible enough to handle the wide variety of data and pattern types found in modern spatial databases because computational methods are designed for mathematical patterns. In contrast, studies such as Anscombe (1973), Guo et al. (2005), Gahegan et al. (2001), and Keim et al. (2005) have shown that there are advantages specific to the visualization approach: uncompressed hypothesis space, ability to process both qualitative and quantitative data, and the inclusion of the human cognition system which is less influenced by data noise. However, neither using a pure computational or visual approach is as effective as the combined use of both approaches.

Geovisualization attempts to visually stimulate and engage the human cognitive function to find patterns. However, geovisualization has traditionally been developed for uni- and bivariate spatial data. The most common medium to present spatial data is a map. Maps allow data to be plotted based on spatial location, which is critical, since concepts of location and space are vital to understanding geographic patterns. Colour brightness and saturation are used to represent quantitative differences between plotted data points. However, colour is only one visual component and thus is mostly suitable for one attribute.

With the growing complexity of spatial databases, there is a lack of capability to fully utilize these new complex spatial datasets due to the fact that current visualization methods cannot properly handle multivariate spatial data. Thus, the main research question addressed in this work is: *how can spatial multivariate data be visualized using an icon based non-fused co-visualization approach?*

There has been a substantial amount of work on multivariate geovisualization. Two examples are the bristle map approach for vector data (Kim et al., 2013) and the IconMapper System for raster data (Pazner and Zhang, 2004). The IconMapper System replaced each raster cell with an icon to produce an icon map. Icon elements embedded within each icon represent different attributes using visual primitives such as colour and size. However, there were several issues with the program that limited its effectiveness and usability including method design, lack of software features, and portability.

1.2 Objective

The objective of this research was to design and implement a new visualization software system based on the older IconMapper System in order to answer the thesis research question. The new visualization software, GeoIcon Viewer, aims to improve the older IconMapper System by:

1. Implementation of a new icon design to improve visual performance.
2. Implementation of new significantly different program features to improve the icon visualization method and the visual analysis process.
3. The removal of dependence and reliance on third party software and operating systems.

An example of a GeoIon Viewer screen is shown in Figure 1.1. The figure shows the graphic user interface of the GeoIcon Viewer which is composed of three windows that display the overview map, legend, and GeoIcon Image Map visualization result. In this example, three attributes are rendered and displayed at each location in the icon image map.

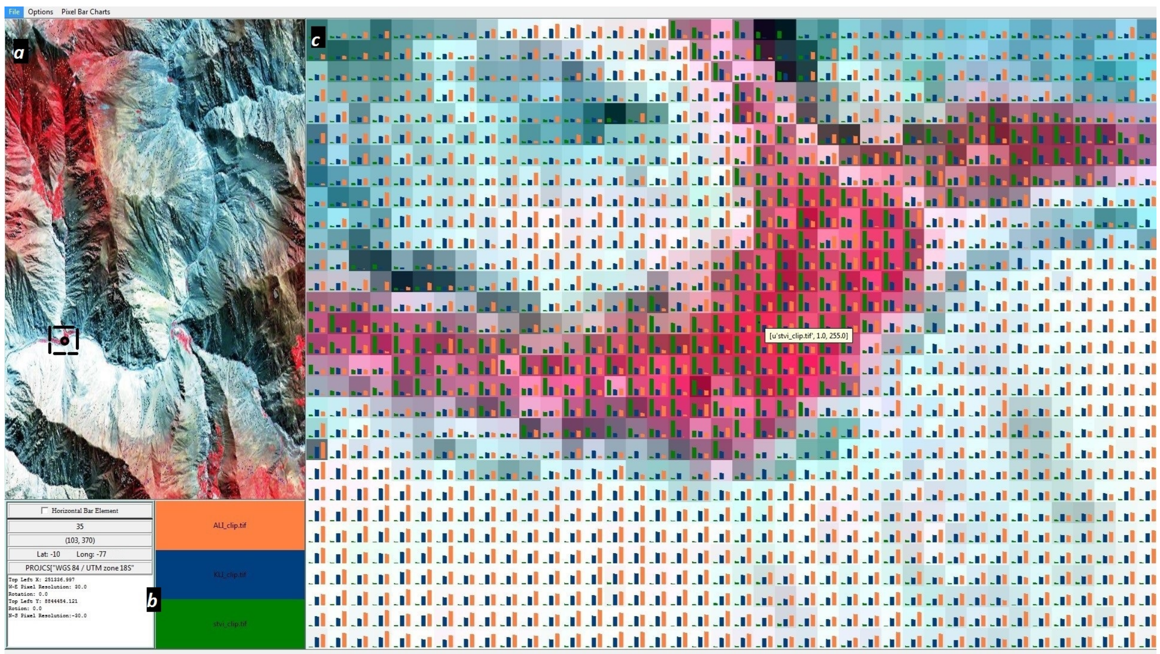

Figure 1.1: Icon image map visualization of alunite, kaolinite, and vegetation. The GeoIcon Viewer user interface consists of the following three windows: a) Overview Window, b) Control/Legend Window, and c) Visualization Window.

1.3 Thesis Organization

Following this introduction, Chapter 2 provides background information on data mining and Knowledge Discovery from Database. The chapter addresses how spatial data require a specialized version of KDD due to inherent data characteristics. The chapter also explores the different stages of (G)KDD and focuses on the visualization approach including current and new methods in spatial data visualization.

Chapter 3 is about the design process of the GeoIcon Viewer. More specifically, the chapter details the transformation process from specific feature requirements into software components. The three main feature requirements are the new icon design (GeoIcon Image Map), data query system, and Region-of-Interest Image Layers Chart visualization method. The design process is approached from three different aspects: input data, visualization methods, and interaction techniques. Designs for each individual program component and the overall system were created based on these three aspects and the specified feature requirements.

Chapter 4 provides a detailed description of the implementation of the software components, as well as how these components interact with each other. A specific programming paradigm known as Object Oriented Programming was used to implement the GeoIcon Viewer based on the programming language chosen. The chapter provides details about the functions and data structures implemented for the major program component such as the visualization processor.

Chapter 5 demonstrates and evaluates the GeoIcon Viewer using a case study involving remote sensing digital image analysis of a copper deposit. GeoIcon Image Map and Region-of-Interest Image Layers Chart outputs are created using ASTER satellite data acquired over the area of the copper deposit. The program evaluation is divided into two parts. The first part examines the GeoIcon Viewer with regard to its desired functions and its actual capabilities. The latter part evaluates the effectiveness of the GeoIcon Viewer and its two visualization methods.

The final chapter presents a summary of the research project and discusses possible additions and changes to the GeoIcon Viewer to improve its role in visual spatial data mining within the context of the GKDD process.

Chapter 2

Literature and Background

The goal of this project is to create a new visualization program. The research field of Knowledge Discovery from Databases helped to form the rationale for the GeoIcon Viewer, and is presented in the following sections.

2.1 Knowledge Discovery from Databases

Knowledge Discovery from Databases (KDD) is developed based on the belief that very large databases contain information in the form of patterns, and is a process for extracting potential information from very large datasets (Miller and Han, 2009). KDD is commonly referred to as data mining. However, data mining is only one component of the KDD process. KDD is suitable for very large datasets with millions of data records which are too complex to be analyzed by conventional statistical data analysis. The purpose of KDD is to uncover hidden patterns, relations, and trends that are obscured by the complexity of the dataset. KDD follows an inductive approach where users examine the results to gain understanding without any *a priori* assumptions and hypotheses about what knowledge will be obtained at the end (Lo and Yeung, 2007) (Miller and Han, 2009). Thus, KDD is considered an exploratory and probabilistic analysis method since it only seeks to uncover rather than to explain hidden patterns.

The Knowledge Discovery from Databases process is composed of multiple stages: data integration/cleaning, data selection/transformation, data mining, knowledge discovery/construction, and lastly deployment (Fayyad et al., 1996) (Miller and Han, 2009) (Gahegan et al., 2001) (Lo and Yeung, 2007). The data integration/cleaning stage consists of finding and combining multiple data sources into one, and problems of missing or erroneous data are rectified. Relevant data are formated for the next step during the transformation stage. The formatting process includes reclassification, denormalization, and aggregation (Lo and Yeung, 2007). Computational and visual approaches help to extract hidden patterns from the dataset in the data mining stage. Users construct hypotheses based on the outputs for further analysis and decision making in the last two stages. The focus of this project is on visualization methods used in the data mining stage. Thus the following sections focus on the data mining stage with a particular emphasis on visualization methods and input data. The focus on input data is due to the fact each computational and visualization method is designed for a specific type of data, and the GeoIcon Viewer is specifically designed for spatial data.

2.1.1 Input Data: Non-Spatial and Spatial Data

Knowledge Discovery from Databases is traditionally designed for non-spatial data for uses in fields such as physics, astronomy, business, and biology (Miller and Han, 2009). However, there has been an explosive growth in digital spatial data within the last few years. Advancements in information technology (IT) infrastructures lead to more powerful data collection, processing, and distribution methods. According to

MacEachren and Kraak (2001) 80% of data generated today are spatial in nature. IT advancements also increase the capabilities of Geographical Information System to support decision making, and this progress is marked by more powerful computations, efficient data analysis techniques, and very large and complex spatial databases (Guo et al., 2005) (Guo, 2003). The magnitude and complexity of these datasets allow for more detailed analysis, but present an extraordinary challenge for traditional KDD in terms of analyzing and transforming these data into knowledge, and prompted the growth of a specialized version of KDD called Geographic Knowledge Discovery from Databases (GKDD).

Digital spatial data are the numerical representation of real world features, and can be categorized into objects and phenomena (Lo and Yeung, 2007). Objects are discrete and definite entities with well defined boundaries, and some examples include roads, buildings, or bodies of water. In contrast, phenomena are features such as precipitation or temperature that are distributed across space without discrete borders. Storing and processing these two categories of spatial data require the use of Conceptual Computational Models (Wise, 2002). There are two commonly used data models for spatial data: Vector and Raster data model. Within a Vector model, spatial objects are stored as one of three geometric objects: point, line, or polygon. Corresponding attribute data for each geometric object are stored within an attribute table. For the Raster model, data is stored as a layer which is simply a tessellation filled with numbers. Each raster or vector data element corresponds to a specific spatial location on Earth.

Spatial Data in Knowledge Discovery from Databases

While spatial databases may be equal in size and complexity with non-spatial databases, the inherent properties of spatial data present difficulties that traditional

KDD process cannot address. There is a relationship or interrelatedness between up to four dimensions of geographic data that is not present in non-spatial data: latitude or northing, longitude or easting, elevation, and time. These four dimensions form the framework for all other spatial data attributes (Miller and Han, 2009). Data mining, especially computational, techniques must take into account the inherent geometric and topological measurement framework that spatial data contains (Miller and Han, 2009).

In addition to dependencies and relationships amongst attributes, spatial data such as temperature or precipitation often exhibit spatial dependency and heterogeneity. Spatial dependency is the tendency for attributes in space to be related, and can be summarized by Tobler's First Law of Geography which stated that "things closer together in space are more closely related than distant things" (Tobler, 1970, p. 236). However, this property is not always true and depends on other spatial attributes such as direction or land cover. Spatial heterogeneity refers to the non-stationarity of geographic processes (Miller and Han, 2009) which means geographic processes vary by location and that the global pattern might be completely different from local patterns. Traditional computational mining techniques were constructed based on data independence and stationarity assumptions, and thus are not suited for spatial data (Miller and Han, 2009).

The complexity of spatial data presents a challenge to the standard KDD process. Non-spatial data are easily represented and processed as points in the information space, but many geographic objects contain size, shape, and boundary properties which also affect the specific geographic process. Furthermore, relationships such as distance, direction, and connectivity are critical in understanding geographic patterns, but are hard for traditional KDD to handle and process.

2.2 Data Mining Stage of KDD

2.2.1 The Computational Approach

The data mining stage contains a variety of methods spanning a continuum of computational and visual approaches. Computational methods are designed to take advantage of the computational power and the formalism of statistical inferences to search and uncover patterns in a comprehensive and consistent manner. The main advantage of computational methods is the speed and efficiency of the algorithms. However, these algorithms are only applicable to quantitative data and mathematical patterns. Qualitative data such as opinions are difficult to process without conversion to ordinal, interval, or ratio data by the user. The lack of automated processing for qualitative data means that computational approaches require human level intelligence inputs (Miller and Han, 2009). In addition, the size and dimensionality of these very large databases still pose difficulties for automatic pattern detection despite the increasing computing power. A large hypothesis space, which is all the possible patterns within a dataset, is present due to the size and dimensionality of the dataset. Computational methods narrow down the hypothesis space because the act of choosing a specific algorithm is to assume what pattern exists in a dataset. In doing so, there is an imposition of an *a priori* hypothesis which results in undiscovered patterns. Lastly, as mentioned in the previous section, the inherent characteristics of spatial data pose difficulties for computational data mining. To address the shortcomings of the computational approach, human level intelligence and visualization methods are used in conjunction with computational statistical methods to form an effective KDD process. The following subsections provide a background on the visualization techniques used in KDD and GKDD with a particular focus on spatial data visualization.

2.2.2 The Visualization Approach

Within the KDD process, visual data mining or scientific visualization techniques involve using information systems to facilitate visualization and interaction with the data in order to promote visual thinking (Worboys and Duckham, 2004). Geovisualization is the sub-branch of scientific visualization geared for spatial data, and integrates different methods from scientific visualization, cartography, image analysis, exploratory data analysis, and geographic information systems to create an enivronment for the visual exploration, analysis, synthesis, and presentation of spatial data (MacEachren and Kraak, 2001). By transforming the visualization space to an explorable multi-dimensional space, visualization approaches act as the link between human intelligence and the rigid computational approaches (Hernandez, 2007). In the past, the focus of visualization research has been in fields such as engineering or medical sciences (Hernandez, 2007). However, the amount of spatial data being collected is promoting research in geovisualization. The rapid growth of spatial datasets presents a major challenge in incorporating geovisualization within the GKDD process. The purpose of geovisualization is to exploit human visual cognitive abilities in pattern recognition, ordering, and interpretation of visual cues (Hernandez, 2007) (MacEachren and Kraak, 2001) (Keim et al., 2005) (Fairbairn et al., 2001). The reason is that humans learn more effectively and efficiently within a visual environment rather than a textural or numerical setting (Tufte, 1990) (Tufte, 1997). In a study Anscombe (1973), four datasets with the same statistical profile were generated as shown in Figure 2.1. Although the statistical profile of these four dataset were the same, their graphs were very different as shown in Figure 2.2. This study highlights that visualization techniques can provide different additional insights about data that numerical analysis could not.

In addition, human eyes are remarkably effective at filtering out noise from useful

Number of observations (n) = 11
Mean of the x's ($\bar{x}$) = 9.0
Mean of the y's ($\bar{y}$) = 7.5
Regression coefficient (b_1) of y on x = 0.5
Equation of regression line: $y = 3 + 0.5\,x$
Sum of squares of $x - \bar{x}$ = 110.0
Regression sum of squares = 27.50 (1 d.f.)
Residual sum of squares of y = 13.75 (9 d.f.)
Estimated standard error of b_1 = 0.118
Multiple R^2 = 0.667

Figure 2.1: The statistical profile of four datasets used in Anscombe (1973).

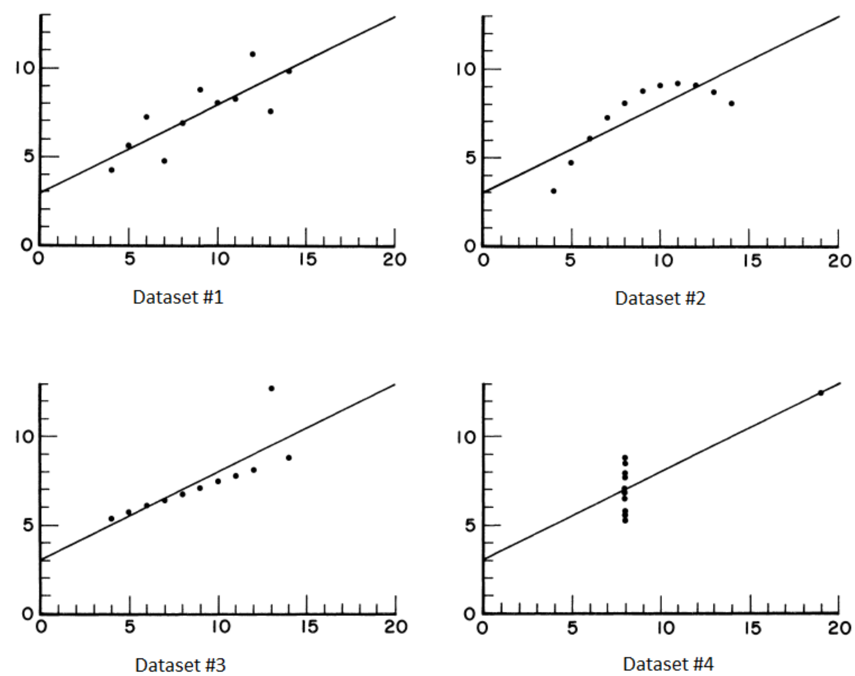

Figure 2.2: Graph of four datasets with the same statistical profile from Anscombe (1973).

data (Fekete et al., 2008). Spence and Garrison (1993) compared the summaries of computational and visualization approaches. Using the Hertzsprung Russell Diagram shown in Figure 2.3a as input data, Spence and Garrison (1993) found that the visual summary in Figure 2.3b was much better than that of any computational approach given the noise within the dataset. The capability of the human visual system in filtering and pattern recognition is what visualization techniques aim to take advantage of.

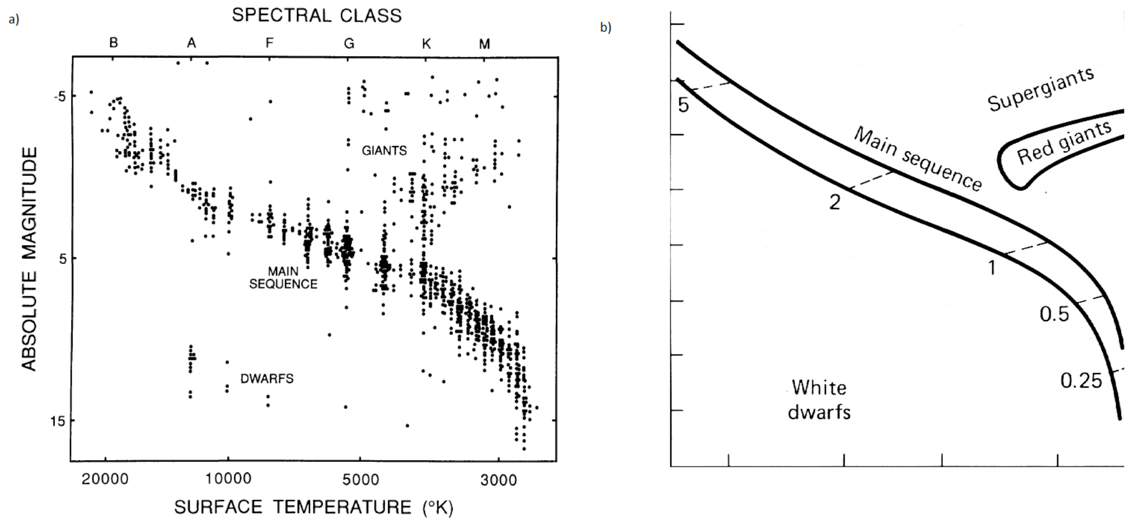

Figure 2.3: a: Hertzsprung Russell Diagram. b: Standard visual summary of the Hertzsprung Russell Diagram. (Spence and Garrison, 1993).

2.2.3 Exploratory Visual Analysis

To incorporate the human cognition system in pattern detection, the KDD process uses an approach known as the Exploratory Visual Analysis (EVA) or Exploratory Spatial Data Analysis (ESDA) whose main goal is to facilitate the exploration and manipulation of data visualizations by users to discover more about the data (Purchase et al., 2008) (Gahegan et al., 2001) (MacEachren and Kraak, 2001). As the name implies, the aim is not to analyze the data, but to present the data in a way that visually stimulates human cognition and intelligence via different data representation styles (Gahegan, 1998).

In a dataset characterized by vast size and high dimensionality, the number of potential patterns is quite high. As stated before, selecting a specific computational method compresses the potential hypothesis space, and may lead to undiscovered patterns. For example, a regression analysis assumes either a linear or nonlinear pattern, and configures the coefficients in relation to that pattern type. In doing so, other possible patterns that might exist are ignored (Guo et al., 2005). One solution is to use multiple algorithms for all pattern types. However, this can take a large of amount of effort for very large datasets, and there is a lack of knowledge as to what pattern types exist. In addition, computational algorithms are not suited for certain data types and patterns without preprocessing, and are also sensitive to noise. In contrast, exploratory visualization facilitates an interactive and unencumbered search for patterns by simply presenting the data to users and allowing the human brain to form insights and conclusions (MacEachren and Kraak, 2001). Visual data exploration can be used for hypothesis generation as well as for verification. EVA plays an important role in the process-pattern tracking where the key aspects of a process are displayed as the process unfolds (Gahegan et al., 2001).

There are several advantages to using a visual exploratory environment in the knowledge construction process. The exploratory environment allows for a more direct involvement on behalf of users. Within an interactive visualization environment, the user can ideally manipulate and visualize data in meaningful ways that maximize the potential of the user's cognitive functions. Also, by using the human brain and cognitive function to observe and form hypothesis about a data set, there is no imposition of *a priori* hypothesis on the dataset, and hence a less likely chance of hidden patterns going undetected. In addition, Spence and Garrison (1993) demonstrated visual data analysis is well suited for non-homogeneous and noisy data. Lastly, using a visual exploratory method requires relatively less understanding of the complex mathematical or statistical algorithms in computational approaches (Keim et al., 2005). However, understanding of different concepts such as data processing, dimension reduction, and visualization method algorithms may maximize the effectiveness of the visual approach.

Different visualization approaches employ different graphical representations of the data, but all visual exploratory methods follow the process of exploration, confirmation, synthesis, and presentation (Gahegan et al., 2001). The sub-steps within the exploration stage are known as the Information Seeking Mantra, and are composed of overview, zoom and filter, and details on demand (Keim et al., 2005). In the overview sub-step, users identify interesting potential patterns. Once a possible pattern is identified, users can visually isolate the data subset and access more details. Users can confirm, construct, and present potential patterns within the dataset through exploration and interaction with data visualizations. Visualization approaches have changed from visualizing the data on a static two dimensional map to within an interactive and dynamic visualization environment. Data are no longer represented by unchanging symbolism on paper, but rather by pixels on a screen based on user specifications. The effectiveness of visualization methods has been growing, and Exploratory Visual Analysis has been used for data exploration in multiple studies such as Keim and Hermann (1998), Keim and Kriegel (1996), MacDougall (1992), Fotheringham and Charlton (1994), Dykes (1997), Hearnshaw (1994), and many more.

A large number of visualization techniques have been developed, and these techniques can be classified based on three criteria shown in Figure 2.4. The three criteria or dimension of the classification system can be assumed to be orthogonal. This means that any visualization technique may be used in conjunction with any interaction technique for any data type to form different visualization environments (Keim et al., 2005).

According to Hinneburg et al. (1999) and Gahegan et al. (2001), common visualization techniques can be categorized into chart-based, pixel-based, glyph/iconographic, and map-based techniques. A chart-based technique plots the data on a graph, and common examples include scatter plots, parallel coordinate plots, and stacked plots. Pixel-based methods map data values to pixels which are then arranged in a specific order. Similar values produce visible clusters in two dimensions, and can provide a

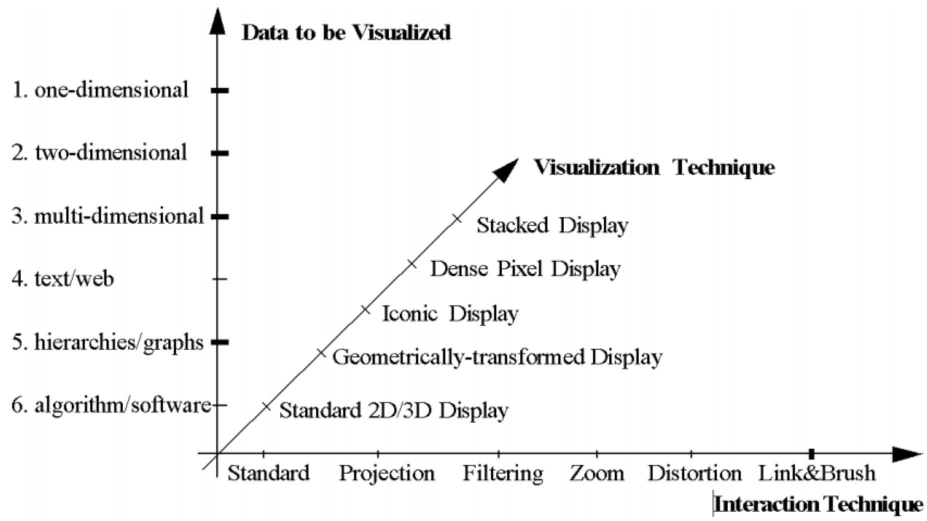

Figure 2.4: Information Visualization Technique Classification from Keim et al. (2005).

useful overview of the dataset. Glyph and iconographic techniques aim to display the perception of 'whole' while allowing differentiation of the individual attribute (Gahegan et al., 2001). The best known example is the Chernoff Face method in Chernoff (1973) where the eyes, ears, mouth, and nose represent different attribute values through shape, size, placement, and orientation. The map-based method is the primary approach in cartography and geovisualization, and is a powerful technique for exploration, extraction, and summary of spatial data (Fairbairn et al., 2001). For data that can be displayed spatially, a digital map representation allows for data exploration but also serves as a foundation for incorporating different representation schemes with varying degrees of abstractions ranging from Chernoff Faces or charts on one end to the Imhof method (Imhof, 1982).

In a map-based approach, geographic coordinates are used to plot data points on a two dimensional cartographic X-Y plane. For the raster model, each data point corresponds to a square within the tessellation. The two dimensional planimetric view is the conventional display method used in geovisualization. The potential spatial patterns can be highlighted or depicted using colours and symbologies (Lo and Yeung, 2007). As computing power increases, the three dimensional perspective view is becoming more common. Geographic coordinates are plotted on two axes (X and Y), while another variable such as elevation is plotted on the third (Z) axis. Depending on the spatial distribution of data points and the map scale, some areas of the display map may contain a higher amount of visual information than other areas. Information density becomes an issue when a large amount of data is plotted in one area on a small scale map. The resulting visualization is a trade off between information density and portions of data displayed. For three dimensional maps, a significant fraction of the data may be obscured unless the view point is changed.

The map-based visualization approach described above is commonly used for univariate or bivariate data. For univariate data, colour is used to show and highlight possible spatial patterns. The purpose of colour is to make information visually distinguishable. A colour may be described by three dimensions: hue, value, and saturation. Hue is the dominant wavelength reflected, and is commonly referred to as the colour. The value or the lightness is how light or dark a colour is with a constant hue. Lastly, saturation is the purity of a hue where a narrower range of reflected wavelength equals a purer saturation Lo and Yeung (2007). In cartography, changes in hue often represent qualitative or nominal changes such as in a thematic map of land type where each colour corresponds to one land type. In contrast, changes in colour value and saturation are used to represent quantitative differences such as in a choropleth map of rainfall amount.

2.3 Multivariate Data Visualization

Past mapping methods have been focused on univariate and bivariate data. The growing complexity of datasets in multivariate mapping continues to be a challenging research topic in geovisualization. The disadvantage of using colour in map-based techniques is the inability to visualize multiple attributes at a single location. The three common approaches to visualizing multivariate data are the fused visualization, non-fused co-visualization, and semi-fused visualization approach. In a fused visualization approach, map algebra operations produce a composite image, which is used to represent and visualize a specific subset of data (Pazner and Zhang, 2004). Attribute data are combined together via mathematical functions such as the Simple Additive Weighting method (SAW) to produce a composite map (Malczweski, 1999). In addition, various overlay methods are used, including translucent overlay.

In contrast, a non-fused co-visualization approach (Figure 1.1) aims to visualize multiple attributes at each location on a single image or map. This type of visualization depicts each attribute independently through different display properties such as geometry, colour, texture, or pattern, and then integrates all of the depictions into one map via methods such as glyphs or icons (Guo et al., 2006). The non-fused co-visualization approach may also incorporate the fused approach where a composite image showing global pattern can be displayed and compared with multiple individual attributes simultaneously.

The **constant format** or **small multiples** method can be considered as a combination of fused visualization and non-fused co-visualization (Tufte, 1990). The small multiples method is where “the same basemap is used in a series but the data visualized change over time” (Kimerling et al., 2012, p. 138). The advantage of this method is that the location or geographic extent remains constant and this allows users to focus on the changes in data.

There has been an increased focus on multivariate data visualization, and many methods have been developed. For non-spatial data, there are techniques such as the parallel coordinate plots (Inselberg, 1985) or the hierarchical pixel bar charts (Keim et al., 2002). For spatial data, relatively new techniques include the IconMapper System (Pazner and Zhang, 2004) and bristle map method (Kim et al., 2013). A bristle map consists of a series of lines extending from linear map elements such as roads, train, or power lines. These lines encode multiple attributes through variations in length, density, color, orientation, and transparency of the bristles. A bristle map for day and night burglary rates is shown in Figure 2.5. The colour of the bristles indicates time of burglary, and the bristle density and length encode the rate of burglary occurrences.

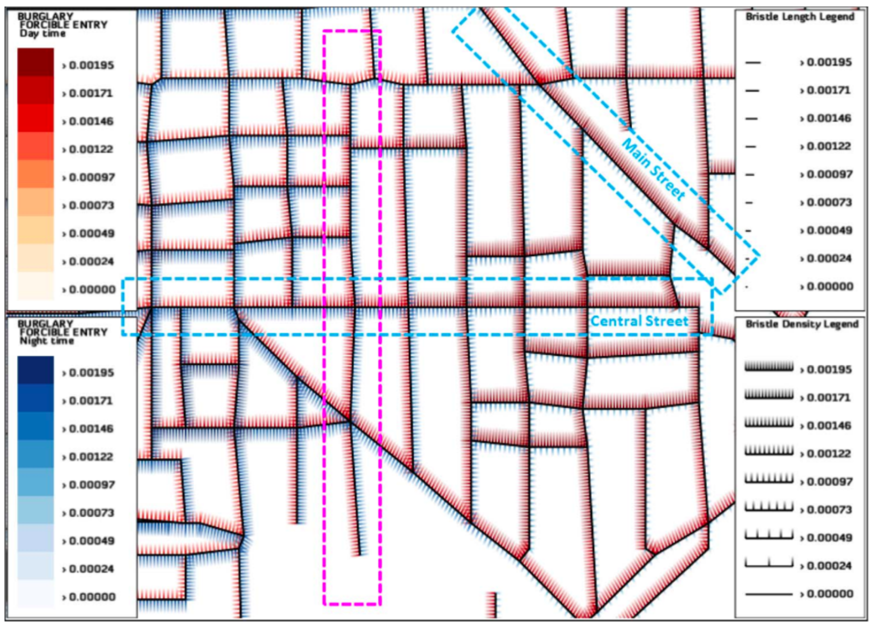

Figure 2.5: Colour, bristle density, and bristle length are used to encode multivariate data about urban burglary rate (Kim et al., 2013).

Unlike the bristle map, the IconMapper System by Pazner and Zhang (2004) is designed for raster data. Miller (1956) found that providing and organizing visual stimulants into multiple dimensions increases the amount of information the brain is able to receive, process, and remember. The IconMapper visualization technique renders an icon at each cell to form more graphically rich and meaningful visualization primitives by using colour, length, width, orientation, shape, and texture to encode multiple attributes. The IconMapper System used a static pictorial and dynamic bar-chart icon template as shown in Figure 2.6.

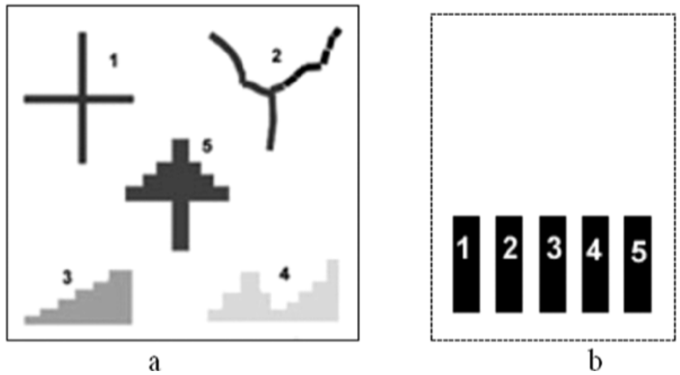

Figure 2.6: Icon design template a) Static pictorial, b) Dynamic bar-chart icon. (Pazner and Zhang, 2004)

A static pictorial template design used five static icon elements to encode five spatial variables as shown in Figure 2.6a. The color of each icon element changed based on the data value. The collective shape and colour of each static icon element indicated the traversability at each location (Pazner and Zhang, 2004). The icon design in Figure 2.6b is a dynamic icon where the size of each bar icon element changes with the value. The color coding of the bar icon elements allows users to differentiate the attributes on display (Pazner and Zhang, 2004). There are several issues with the IconMapper System visualization technique, some of which are listed below.

1. As shown in Figure 2.7, the display of a large number of icons results in high visual information density and hinders the visual exploratory process. This hindrance is further exacerbated by the process to interpret icon element colour and size.

2. The design of the dynamic bar icon leads to misleading visual effects such as the seemingly continuous vertical line across multiple cells as shown in Figure 2.7.

3. The IconMapper System was able to visualize five attributes at each location.

However, this might not be sufficient for high dimensional databases used in modern GIS.

4. Users cannot customize the icon element colour which may lead to unintuitive colour representations.

5. The IconMapper query system returned the date of creation, attribute layer used, and the style of icon used. However, this information does not contribute much to exploratory visual analysis.

As a result of these design issues, the IconMapper System did not serve as an interactive and effective visual exploratory program. The IconMapper System processed the input raster to produce an icon map, but did not allow users to select and focus on a particular region of interest (ROI). An interactive program, ideally, allows users to select, create, and explore data visualizations in real time rather than waiting for a static image file to be created and viewed.

A high dimensional database may contain ordinal, interval, and ratio attribute data, and the range for each attribute may differ significantly. A standardization procedure is required for the different attributes. Due to the number of display pixels allocated to each icon, the visualization of data value by each icon element is very coarse as can be seen in Figure 2.7. Similar data values result in bar icon elements that are indistinguishable in size, which makes it harder to gain an accurate understanding of the data. As the number of pixels allocated to each icon element is dependent on the number of attributes being visualized, more attributes result in icon elements with negligible size differences. Thus, a visualization method is needed to display a high number of spatial attributes simultaneously in such a way that relatively small value differences can be displayed and differentiated. A potential solution is the hierarchical pixel bar chart by Keim and Hermann (1998). The basic idea of a pixel bar chart is

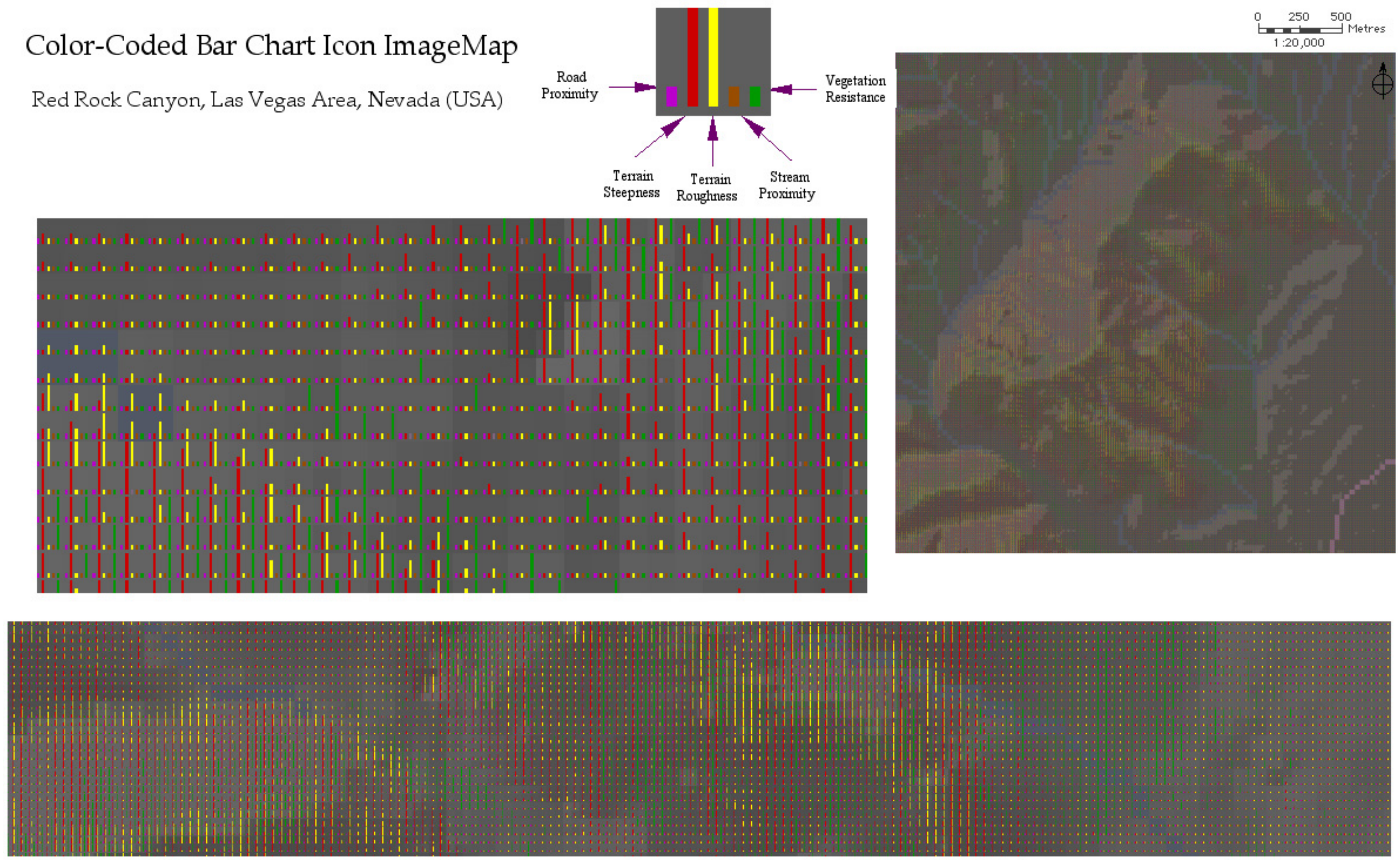

Figure 2.7: IconMapper System dynamic bar icon visualization output. As the image area increased in size, more and more icon elements are rendered, and creating higher information density. Image taken from (Pazner and Zhang, 2004).

to represent each data item by a single pixel within the bar chart, and the data value is encoded by the pixel colour and can be accessed as needed.

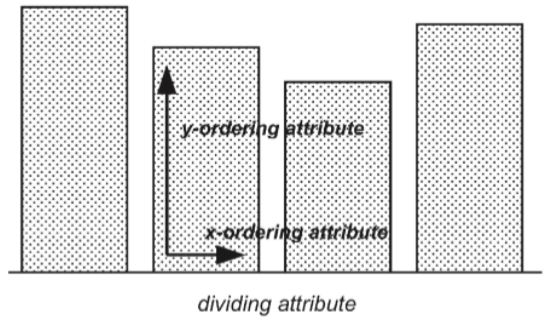

Figure 2.8: Pixel bar chart design (Keim et al., 2002).

The data is arranged in a particular order within each bar using the X and Y ordering attributes as shown in Figure 2.8. The dividing attribute is used to separate the entire input data into multiple pixel bars. The pixel bar chart shown in Figure 2.9 displays multiple attributes simultaneously by using the product type as the dividing attribute while the number of visits and money spent are the X and Y ordering attributes respectively. The colour of the pixel encodes the amount of money a customer has spent.

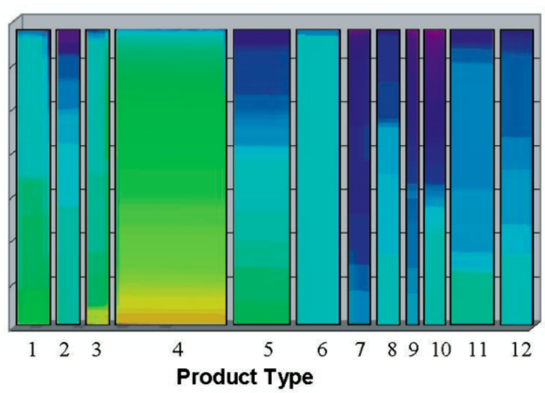

Figure 2.9: Pixel bar chart for consumer information (Keim et al., 2002).

The hierarchical pixel bar chart and the small multiples approach are combined to create a visualization technique for multivariate spatial data with negligible value differences. The design and implementation of this new visualization method, Region-of-Interest Image Layers Chart, are presented in the following chapters.

2.4 Summary

Due to the growing complexity of spatial datasets, new techniques are required to uncover hidden patterns and information from these large and complex datasets. Visualization methods provide a bridge between computational outputs and the human cognition system within the (G)KDD process. Visualization methods try to uncover hidden patterns within large datasets by presenting data in visually stimulating forms to engage the human cognition system. Visualization methods such as the IconMapper System provide an environment for exploratory visual spatial data analysis, but there are issues. The icon design used in IconMapper System was only capable of visualizing five attributes simultaneously. Furthermore, the icon design lacked a dynamic positioning system for the icon elements which resulted in an inefficient use of icon pixel space. The design of the icon elements also created unwanted visual effects such as vertical bar stacking as shown in Figure 2.7. The colour of each icon element was predefined based on position, and this leads to an unintuitive colour representation. In addition to the inefficient icon design, the IconMapper System lacked an effective data query system. At the beginning of the next chapter, the design for a new software is provided which addresses all the design issues of the IconMapper System, and whose development is the focus of this thesis.

Chapter 3

System Design

3.1 Introduction

The goal of this project was to create new multivariate spatial data visualization software based on the IconMapper System by Pazner and Zhang (2004). The GeoIcon Viewer features a new icon design, data query system, and the Region-of-Interest Image Layers Chart method. The end goal of the new visualization software is to provide a dynamic and interactive environment for effective visual exploratory analysis. The entire development process of the GeoIcon Viewer follows a Waterfall development model shown in Figure 3.1.

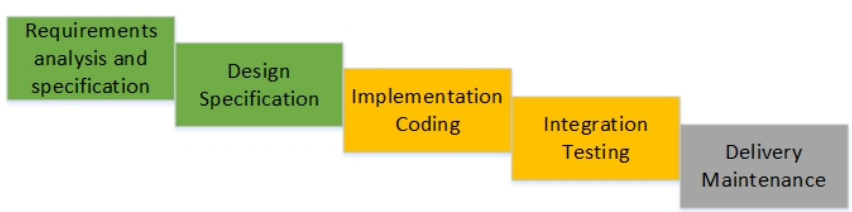

Figure 3.1: The GeoIcon Viewer development process follows the traditional Waterfall model approach.

This chapter focuses on the first two stages, highlighted in green in Figure 3.1, of the Waterfall model. The design stage is critical and a prerequisite for later stages of software development. The design and implementation of the GeoIcon Viewer follows a reductionist philosophy where the final visualization software is the sum of its components and processes. However, it is the purpose of the software that dictates the number and type of components necessary. A clear understanding of the overall purpose and functionality of the software is necessary to create an effective design. A more detailed design process is shown in Figure 3.2 where feature requirements are specified and used to guide the design process.

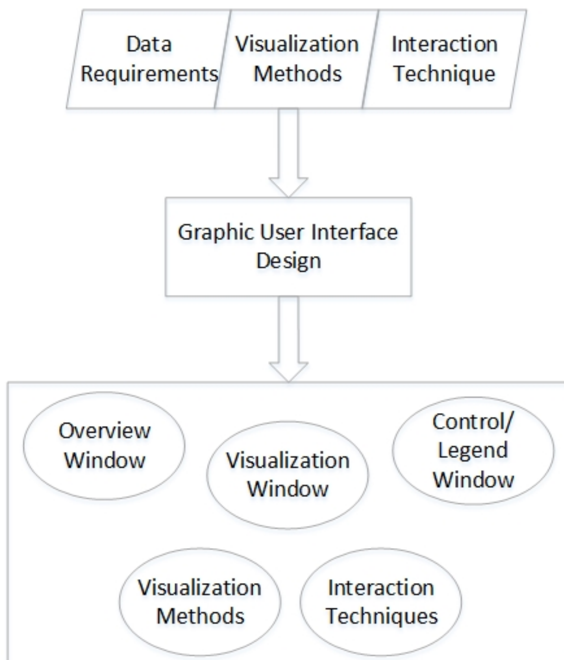

Figure 3.2: Feature requirements are used to guide the design process and specify required program components.

3.2 Overall System Design

The purpose of system design stage is to examine and specify the purpose and required functionalities of the GeoIcon Viewer. Certain functionality requirements have to be met in order for the program to fulfill its goal. Based on the classification system shown in Figure 2.4, there are three design criteria for a visualization program that must be considered: input data, visualization methods, and interaction techniques. These three criteria help to design a suitable graphic user interface (GUI) as shown in Figure 3.2. The GUI is broken down further into several subcomponents where each requires further design and implementation.

3.2.1 Input Data Requirement

The GeoIcon Viewer is most suitable for visualizing raster data due to the inherent characteristics of the iconic visualization approach which works by replacing each raster cell with an icon that displays multiple attributes through a combination of visual primitives. While it is feasible to create an icon object for vector data, there are better visualization methods such as the bristle map approach by Kim et al. (2013). The Region-of-Interest Image Layers Chart method shown in Figure 2.8 is also well suited for the raster data structure, as each raster cell is easily mapped to a set of pixels within each bar. Thus, the visualization methods dictated that the GeoIcon Viewer must be able to handle and process data in raster format.

Raster File Formats

Raster data can be stored in several different file formats and each format has different characteristics depending on the data source or the compression method (Lo and Yeung, 2007). The different formats include generic raster file, raster data interchange, and proprietary GIS software format. Generic raster format includes ASCII and binary raster files, which are simply a list of cell values and are exchangeable between different platforms. However, the raster data interchange format has largely replaced the generic raster format (Lo and Yeung, 2007). The *Tagged Image File For-*

mat or TIFF is the most common raster data interchange format. TIFF is platform independent and supported by a wide range of GIS hardware and software. *GeoTIFF* is able to store spatial information including map projections and georeference coefficients to support spatial operations. TIFF also supports data compression which allows efficient storage of large raster datasets. Proprietary formats such as .GRID or .PIX are created and can only be opened by specialized GIS software such as ESRI ArcInfo or PCI Geomatica.

TIFF is the logical choice for raster data format because it supports the storage of spatial information and is platform independent. Having the ability to handle and process TIFF files fulfills the input data requirement. The ability to retrieve spatial information stored in a TIFF file is essential since the GeoIcon Viewer is focused on spatial data visualization.

3.2.2 Visualization Methods

The chosen visualization method is the second criteria to consider. Since the goal is to use the GeoIcon Image Map and Region-to-Interest Image Layers Chart methods to visualize multivariate spatial data, the GeoIcon Viewer must be able to create and display these two objects. The required input data are the raw values from GeoTIFF images. Thus, the program must be able to retrieve a subset or all of the raster data values. The program manipulates and processes the raw data to create visualization outputs, which are then displayed in a separate GUI window.

3.2.3 Interaction Techniques

In addition to the input data and visualization methods, an effective visual exploratory analysis environment requires interaction techniques to allow users to explore, interact, and manipulate visualization outputs as needed. The two interaction techniques required are View Enhancement and Selection. View Enhancement methods allow users to adjust the level of detail being visualized. The Selection techniques provide users with the ability to isolate a subset of the displayed data, and are the basis for operations such as highlighting, filtering, or additional quantitative analysis methods.

A well known View Enhancement technique is zooming. Working with large datasets requires the ability to present the data in a compressed overview format as well as in a variety of higher resolutions. An icon object becomes larger in size but also presents more details when zoom is applied. There are two common ways to incorporate the zoom feature. The first is to apply the zoom to the entire dataset and replace the original image in the display window with the zoomed image. The second is by having multiple display windows where each window corresponds to a specific zoom level. Comparison of the two methods is difficult as there are unique advantages to each design. A single display window design is space efficient and more appropriate for mobile displays which have smaller screen resolutions and physical dimensions. In contrast, a multiple window design is able to display data at different detail levels simultaneously but is more suited for large monitors, and is the reason why the GeoIcon Viewer uses a multiple window design.

The Selection technique is critical to the overall function of the program and in particular the data query feature. The View Enhancement technique influences how the Selection method is designed. The brushing and linking technique is the required Selection approach for a multiple window design GUI. The idea of brushing and linking is to use multiple visualization methods to overcome any disadvantages of a single method (Keim et al., 2005). Brushing is the act of dynamically querying information related to a selected data object. Linking means that changes to selected data objects in one display window are reflected by changes in the other windows.

The GeoIcon Viewer uses two types of interaction technique: View Enhancement and Selection. Zoom is implemented via a multiple window GUI design, where each window displays the data at a specific scale or zoom level. The brushing and linking technique, which is required due to the multiple window design and the selected visualization methods, enables user exploration and interaction with visualization outputs.

3.2.4 Resulting GUI Design

Feature design requirements were drafted based on the three criteria described above. The GeoIcon Viewer must be able to read, process, and extract pixel data and spatial information from TIFF files. Secondly, the program must be able to generate GeoIcon Image Map and Region-of-Interest Image Layers Chart outputs, which are the icon image map and image layers chart respectively. Thirdly, two types of interaction methods are required: zoom and selection. The zoom feature design led to the multiple display windows GUI design, which in turn dictated the use of the brushing and linking method as the selection technique.

The conceptual GUI based on these design requirements is shown in Figure 3.3. The GUI is comprised of the Overview, Visualization, and Control/Legend windows. These windows are linked so that changes in one window are reflected in the other windows. Once the system design requirements are listed and the GUI is conceptualized, the design process moves onto the second stage of the Waterfall model and focuses on the design of individual program component including the GUI windows and the visualization processors.

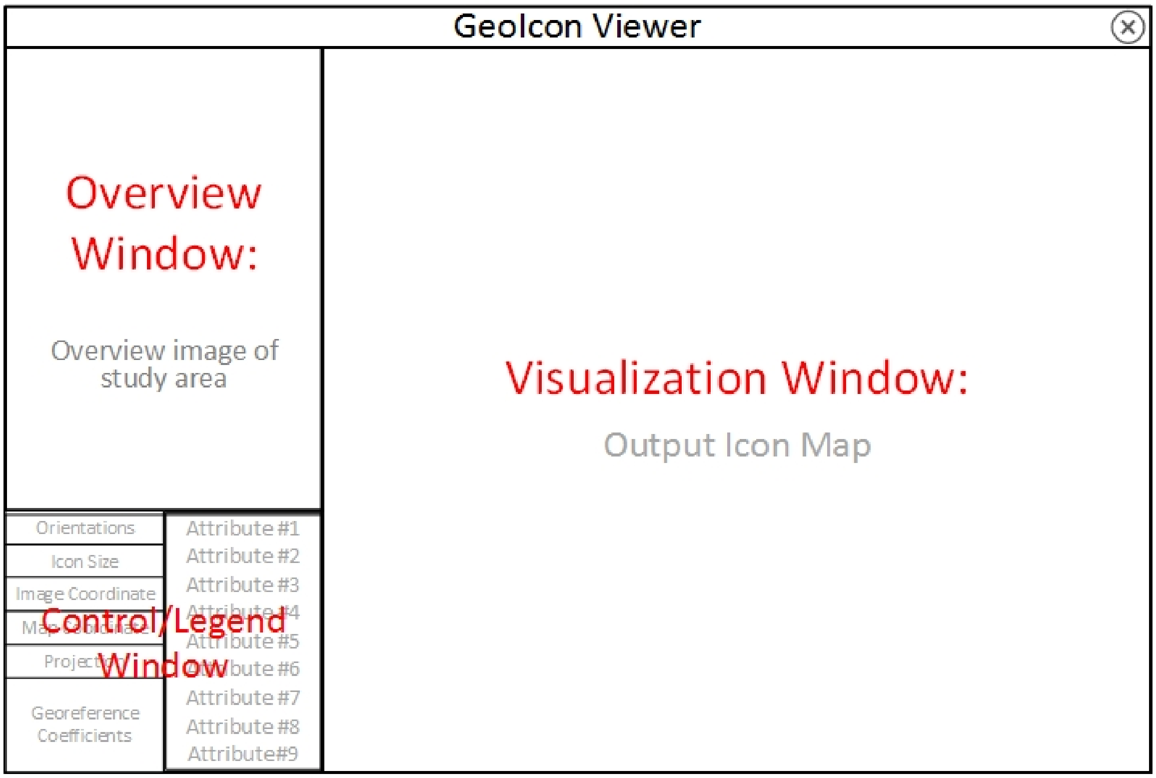

Figure 3.3: The GUI is made of up three windows, each of which has a specific purpose. The GUI design is influenced by the interaction techniques selected.

3.3 Components Design and Specifications

This section deals with the function and design of each individual program component shown in the bottom rectangular box of Figure 3.2. Each component was designed for a specific purpose and combines together to form the GeoIcon Viewer and all its required functions following a reductionist approach.

Overview Window

The Overview Window provides a basis for interaction techniques and displays the overview image of the study area. Users select a region of interest (ROI) on the overview map via mouse click. This bring up more detailed information within the other two GUI windows. The user selection isolates a specific data subset referenced by the spatial or image coordinates. The Overview Window also provides a frame of reference for users during the exploratory analysis. When users select a specific region on the overview map, it is highlighted to provide a link between the visualization outputs and the actual spatial location.

When users select a raster image, it is displayed in the Overview Window. For raster data, each attribute is stored as a grayscale TIFF image where the number within each cell is the attribute value. When a grayscale image is used as the overview map, the chosen image might not be suitable for display. For example, if a chosen image has a histogram similar to the one shown in Figure 3.4a, the displayed image contains very low contrast and is useless for visual examination. In order for users to distinguish any details, prior image enhancement has to be performed to produce a more meaningful overview image.

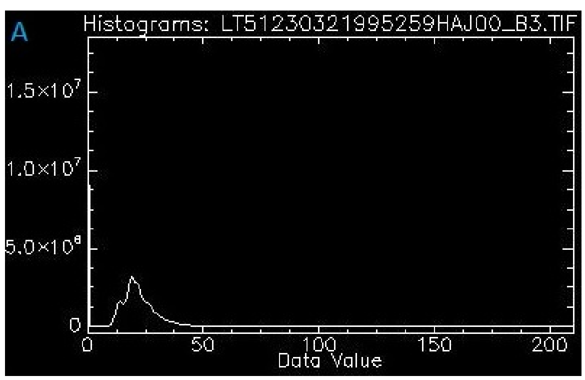

Figure 3.4: Histogram profile of Band 3 image of Landsat 8, where the majority of values are grouped around 10 to 50 and thus the image contains no contrast.

Using a preprocessed grayscale image provides more information to users. However, a grayscale image is still lacking compared to a colour image in visual stimulation and interpretation. As shown in Figure 3.5, the preprocessed grayscale image allows for more details to be distinguished, but the colour image allows for an even easier interpretation and identification of possible land use types. Thus, the GeoIcon Viewer allows users to display colour images as well as creating false colour composite (FCC) for the overview image. In some cases, users might not have sufficient remote sensing data to create a FCC nor a colour image of the study area, and thus the GeoIcon Viewer also supports the display of grayscale images. The program automatically performs histogram equalization to provide a more meaningful display image.

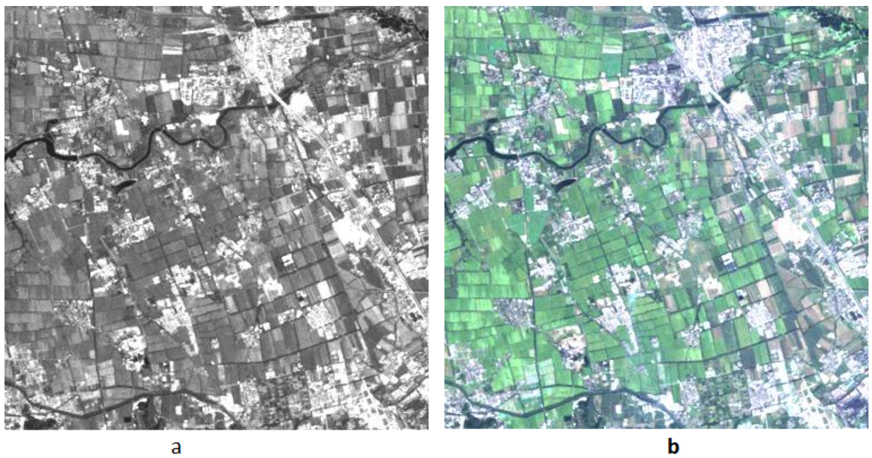

Figure 3.5: a: An histogram equalized grayscale image. b: A true color composite of same area.

In summary, the Overview Window is the first component of the graphic user interface and addresses the input data requirement. In order to handle raster data and provide an effective visual display, the Overview Window supports grayscale, colour, and user created false colour composite TIFF images. The Overview Window also provides the environment or facility for user interaction including selection and zoom.

Visualization Window

The Visualization Window (VW) shown in Figure 3.3 is the second GUI component and displays the icon image map. The Visualization Window is linked to the other two GUI windows. User selections in the Overview Window provide the input data for the visualization processors. The query feature in the IconMapper System retrieved meta-data information such as the visualization method used or the date of creation. Selections of icon elements in the Visualization Window via mouse clicks trigger the new data query process which returns the input data values and attribute name.

The Region-of-Interest Image Layers Chart output or image layers chart is displayed in a pop-up window rather than in the Visualization Window. The reason for this design is to allow users to view both visualization outputs concurrently. The main visualization method in GeoIcon Viewer is the GeoIcon Image Map technique. The purpose of Region-of-Interest Image Layers Chart is to support the GeoIcon Image Map method when the sizes of the icon elements are hard to distinguish. The Region-of-Interest Image Layers Chart display window is linked to the three main GUI windows and also supports data query. The design of the Region-of-Interest Image Layers Chart method is described later on in this chapter.

Control and Legend Window

The Control and Legend Window (CLW) in Figure 3.3 does not display any visualization outputs but rather provides users with additional information about the current region of interest and visualization output properties, including the current location, size of the icon, and the spatial coordinate system. The spatial coordinates allow users to incorporate other data sources such as high resolution satellite images for a more effective visual exploration. The legend displays and changes with the selected attributes and their corresponding colours. The legend works in conjunction with the data query system embedded in the VW for fast identification of icon elements.

3.3.1 Icon Design

The IconMapper System used the icon design shown in Figure 3.6a. In that design, the height of each bar icon element changed with the data value. The color coding of the bar icon elements allowed users to differentiate the attributes on display, but these colours were fixed (Pazner and Zhang, 2004). The icon element size changed in the vertical dimension, but its horizontal dimension was fixed. The position and the width of each icon element remained constant regardless of the number of attributes on display. Furthermore, the maximum height of each icon element extended to the edge of the icon. These design choices together led to the unwanted visual effects shown in Figure 2.7.

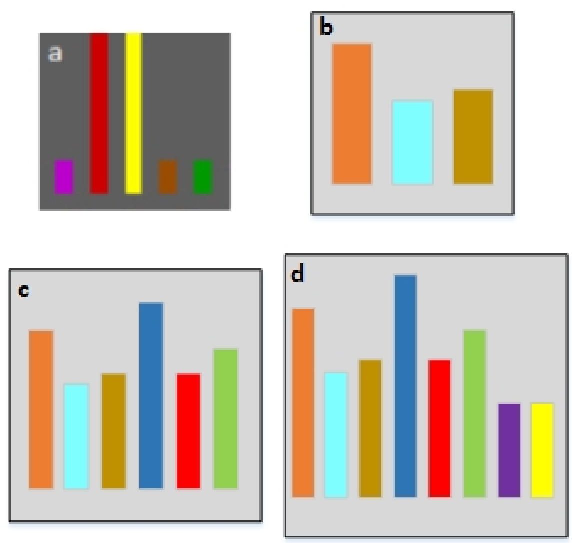

Figure 3.6: a: IconMapper System icon design. b: New three or less attributes icon design used in GeoIcon Viewer. c: New four to six attributes icon design. d: New seven to nine attributes icon design.

Size

The new icon design shown in Figure 3.6b-d aim to address and improve on the design issues mentioned above. The icon size is dependent on the number of attributes users have selected. The more attributes selected, the bigger the icon size. One reason is to allow sufficient icon size so that users can distinguish the differences in icon element size. More importantly, a dynamic or variable icon size enables the visualization of more attributes within each icon. As stated in Chapter 2, spatial databases are growing in dimensionality and size. Having the ability to visualize more than five attributes simultaneously allows for more complex visual pattern detection. The original program requirement was to visualize up to ten attributes due to the fact composite index images and models often incorporate a similar number of variables. However, a square icon was better suited for nine attributes arranged in a three by three configuration (which ultimately was not implemented).

There are three icon sizes for three ranges of attribute numbers. For three or less attributes as shown in Figure 3.6b, the icon size is 35 by 35 pixels. For between four to six attributes in Figure 3.6c, the icon size is 50 by 50 pixels. Lastly, the output icon size is 70 by 70 pixels for seven to nine attributes as shown in Figure 3.6d. The change in icon size allows more attributes to be visualized while at the same time maintaining readability.

Besides the number of attributes selected, the size of an icon is also affected by the graphic user interface. The number of icons that can be displayed is lower as the display window (VW) is smaller in a multi-window design compared to that of a single window design. There is a trade-off between the icon size and the number of icons displayed. Larger icon allows smaller value differences to be distinguished. However, a smaller number of icons can be rendered at once. In contrast, the IconMapper System output shown in Figure 2.7 had too many icons and resulted in non-visually well resolvable high information density. In particular, icon element sizes were difficult to visually resolve and compare. Limiting the spatial extent of the output map to just the neighbourhood of the user selected location reduces the number of icons displayed, but increases the resolution of each icon. The decision on the number and size of icons is subjective, and is made by the developer but ideally should follow user testing and feedback.

The width and position of the icon elements changes depending on the number of attributes as shown in Figure 3.6b-d. This change means that the icon pixel space is more optimally utilized no matter how many attributes are displayed. In addition, the height of each icon element is limited. There is a predefined border space around icon elements in order to mitigate the vertical bar effect shown in Figure 2.7.

Colour

Colour is an important visual component that allows users to identify each icon element. Each icon element is preset to a certain colour during the development stage of the IconMapper System. In the GeoIcon Viewer, users assign a colour to each attribute selected as shown in Figure 3.7. The software automatically queries the directory of the overview image and lists all TIFF image files. Users click on each attribute image name to bring up the colour picker. Allowing users to assign their own choice of colour to each atttribute produces a more intuitive colour representation. However, the trade-off is that multiple colour choices may lead to undesired negative visual effects. Examples include close colour ambiguities or unanticipated problematic colour interaction and patterns.

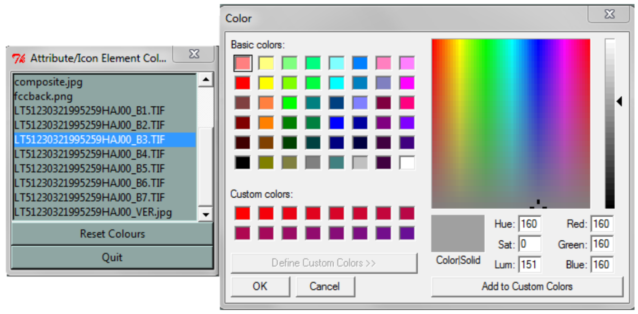

Figure 3.7: Dialog window for customizing the colour of each icon element.

3.3.2 Region-of-Interest Image Layers Chart

The purpose of Region-of-Interest Image Layers Chart method is to act as a visual support aid for the GeoIcon Image Map visualization method. The advantage of the Region-of-Interest Image Layers Chart method is its ability to visualize and highlight negligible data value differences. Data values within a neighbourhood are often very similar based on Tobler's First Law of Geography (Tobler, 1970). The trade-off between the icon resolution/size and the number of icons displayed is unavoidable due to the current display technology, and this means that small value differences are harder to visually distinguish with more icons displayed. For example, a value difference of 0.01 results in an icon element size difference of only 5 pixels if the allocated pixel size for an icon element was 500 pixels. The size difference varies depending on the icon size and the standardized data value. Icon elements may only differ by 1 pixel for value differences in the thousandth place and hence it is very difficult if not impossible for users to visually detect and compare.

The Region-of-Interest Image Layers Chart is a semi-fused visualization method created from a combination of the hierarchical pixel bar chart and the small multiples technique. Each pixel bar or image layer bar corresponds to an attribute the user has selected (Figure 3.8). Each image layer bar covers the same geographic extent which

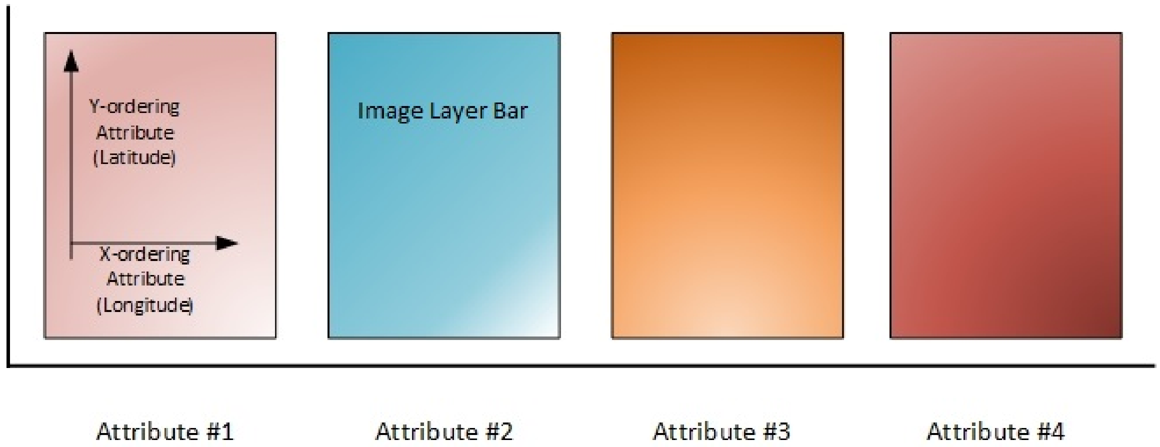

Figure 3.8: Design for the Region-of-Interest Image Layers Chart method.

allows users to focus on spatial patterns. Pixels are arranged by spatial coordinates within each image layer bar as shown in Figure 3.8. In doing so, each pixel within the image layer bar is spatially referenced. The colour of each image layer bar ranges from the user selected colour to white and represents the maximum and minimum standardized value respectively. Small differences in data values are more visually distinguishable when stretched into the three colour channels. The Region-of-Interest Image Layers Chart, like the GeoIcon Image Map method, is capable of visualizing up to nine different attributes simultaneously.

3.3.3 Summary and the Next Step

Chapter 3 described the design process of the GeoIcon Viewer following the Waterfall model shown in Figure 4.1. The first step was to list the design requirements for input data, visualization methods, and interaction techniques. These requirements in turn influenced the graphic user interface design. The GUI itself was broken down into different components, each of which serves a specific purpose in the program. The new icon design aims to improve the shortcoming of the previous design by incorporating dynamic icon element sizes, icon element positioning, customizable colour selections, and lastly an increase in the number of attributes that can be visualized. The Region-of-Interest Image Layers Chart method is designed to support the GeoIcon Image Map method for regions with negligible value differences. The next chapter moves on to the third and fourth stages in the Waterfall model, and focuses on the implementation of all the different program components discussed in Chapter 3.

Chapter 4

Software Implementation

4.1 Implementation Process

The Software Implementation chapter is about the third and fourth stage of the Waterfall model shown in Figure 4.1. The design stage specified a number of program components that needed to be implemented, including a TIFF file handler, graphic user interface, query system, and visualization processors. Each program component was developed and tested independently and integrated into the program. This chapter is divided into several sub-sections including programming paradigm, data flow, component implementation, and integration.

Several terminologies are defined below to make the implementation process easier to follow. GeoIcon Image Map is the name of the visualization method using the new icon design, and this method produces an icon image map. The Region-of-Interest Image Layers Chart method generates an image layers chart which is composed of image layer bars.

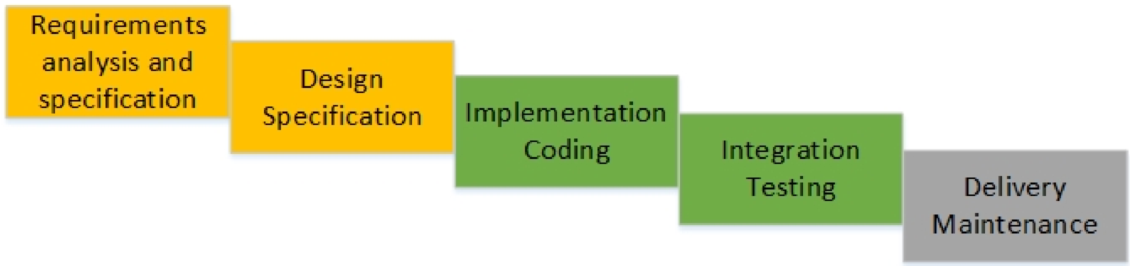

Figure 4.1: Once design requirements are specified, the next steps are implementation and integration.

4.2 Implementation: Program Paradigm

The program paradigm is the style of programming used during software development and is dependent on program specifications and the chosen programming language. A feature requirement of the GeoIcon Viewer is portability and platform independence. While programming languages such as C or C++ are platform independent (Lutz, 2013), Python is the programming language chosen to implement the GeoIcon Viewer. The standard implementation of Python compiles and runs on nearly all major platforms in use today including Linux, Unix, Windows, and Mac OS. Python offers several additional advantages over C or C++ in addition to portability and platform independence. The development productivity in Python is higher than C, C++, or Java since Python code is one-third to one-fifth the size of the equivalent code in other languages. Shorter code requires less maintenance and debugging. The downside of Python is that its execution speed may be slower than C or C++, and whether the execution speed is an issue depends on the purpose of the program. Operations such as animation or numeric programming require the execution speed of C or C++. However, the execution speed is less critical for visualization software such as the GeoIcon Viewer. In addition to its high development productivity, Python also has a vast number of libraries or modules to support a wide variety of operations and tasks. NumPy, which is used in the GeoIcon Viewer implementation, is an add-on library and has been described as a free and more powerful equivalent to the MatLab numeric programming system (Lutz, 2013).

The chosen program language influences the programming paradigm as each language is designed to follow specific paradigm/s. Python is an object-oriented language, and its class structure supports Object Oriented Programming (OOP). A Python class is a device used to implement data objects in Python, where each object is an instance of a class. Each instance object inherits the class's attributes and methods, and all members of the same class have the same attributes and methods. For example, the CAR class contains attributes that describe a vehicle including brand name, colour, and price. Any instance of the CAR class contains these three attributes. Furthermore, a new class such as SEDAN can be derived using the CAR class. The SEDAN class inherits all of the attributes in CAR class, but can contain additional attributes specific to the SEDAN class such as interior material, condition of use, or kilometers driven. The idea is to create new classes by customizing a preexisting class rather than starting from scratch, and this process is known as inheritance. The ability to create children or derived classes means code re-usability and a more efficient development process.

4.3 Data Flow

The Data Flow section describes and illustrates how program objects interact with each other. The data flow algorithm shown in Figure 4.2 provides a framework for how each component functions and is integrated within the whole system. Along with the design requirements, the data flow algorithm informs the development of all program

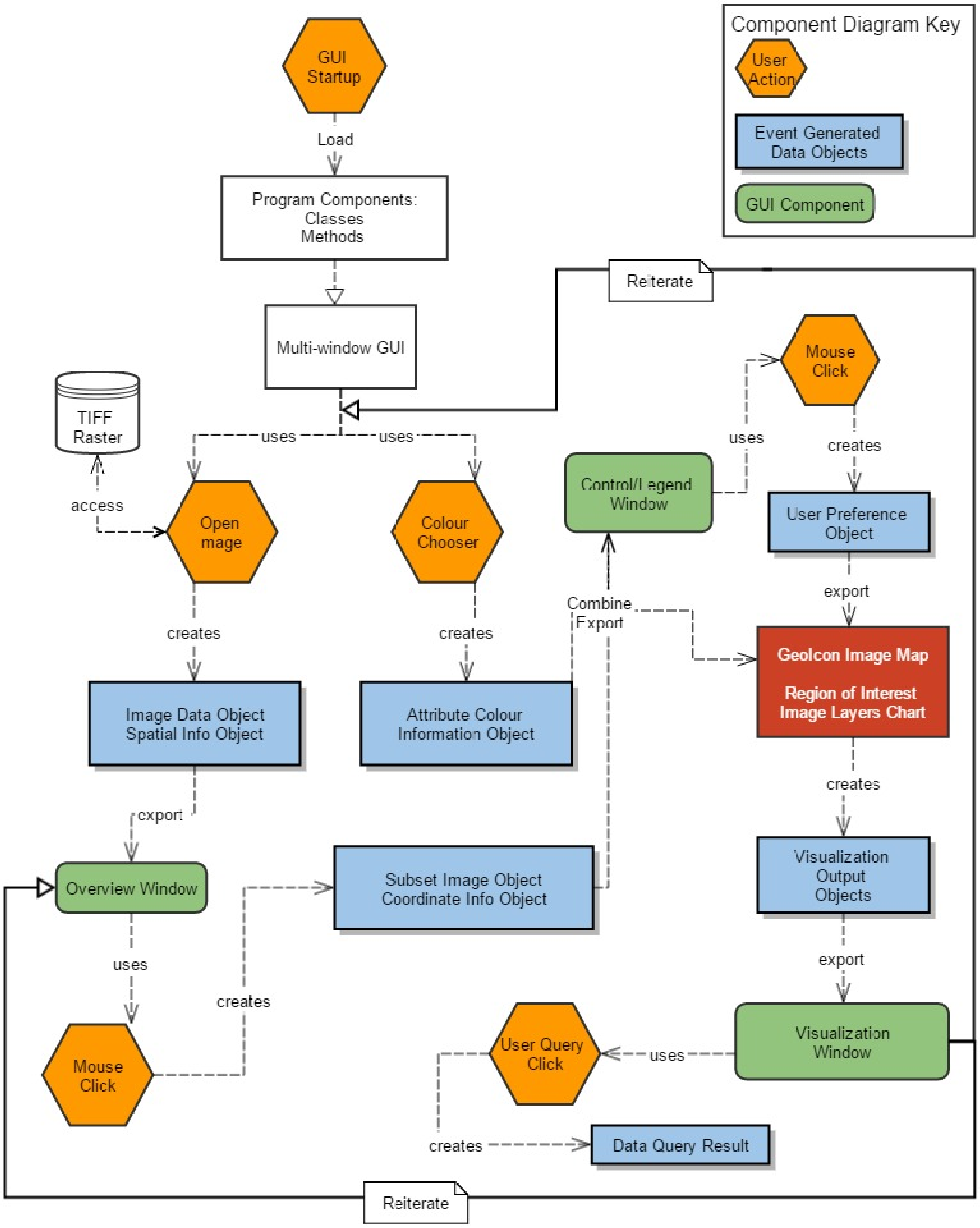

Figure 4.2: The overall algorithm for how different program components interact within the GeoIcon Viewer.

components. The whole process starts with graphic user interface initialization which loads the required Python libraries, classes, methods, and data objects. Once the

GUI has initialized, the program transitions to an event-driven programming state, where the flow of the program is determined by user events such as mouse clicks or key presses. The OOP paradigm is still in use as the program creates and modifies class instance objects in response to user events.

The next step after the GUI has initialized is to import the input data. This process depends on the program's TIFF file handler to retrieve raw data values and the spatial information stored within each image. As stated in the System Design chapter and shown in Figure 4.2, the selected TIFF image is stored as an Image Data Object and passed to the Overview Window (OW) for display and further processing. Within the OW, users select and focus via mouse clicks on a specific location of the display image, which triggers the retrieval, creation, and transfer of two new data objects: Subset Image and Coordinates Info objects. These two data objects are input data for the GeoIcon Image Map and Region-of-Interest Image Layers Chart visualization processors (highlighted red in Figure 4.2). The visualization processors also require an Attribute Colour Information (ACI) object as input. The ACI data object is created by the Colour Chooser handler shown in Figure 3.7 and it contains information about the selected attributes and their corresponding colours. The ACI, Image Subset, and Coordinate Info data objects are required to create an icon image map and an image layers chart.

The GeoIcon Image Map object is passed to and displayed in the Visualization Window. The output icon image map is also used for the implementation of the data query system. Selections via mouse clicks on the icon image map return a Data Query object that contains the attribute name and data value. The program has completed its first iteration once the icon image map has been displayed. At this stage, users may select and focus on another region of the study area via the Overview Window, change the displayed attributes or colours, or change the study area by opening another TIFF image as indicated by the thick black arrow lines in Figure 4.2. The program responds accordingly to different user events.

4.4 Component Development and Implementation

The previous section, Data Flow, described the interaction between user generated events, data objects, and different GUI components within the GeoIcon Viewer. The implementation process of the GeoIcon Viewer is separated into three parts, where each part addresses one of the three GUI windows. Each GUI window contains specific features and design requirements. Separating the entire development and implementation process by the GUI windows means that all of design requirements are fulfilled in a systematic and logical order. The Overview Window focuses on input data handling and processing. The Control/Legend Window and the Visualization Processors focus on the visualization outputs. The Visualization Window is designed for displaying and querying icon image map.

4.4.1 Python Libraries

In addition to the Python standard library, which offers a wide range of operations for everyday programming, there is a growing collection of several thousand library packages which vary in size and complexity from individual module or program to a full application development framework. Three library packages were critical in the development and implementation of the GeoIcon Viewer: Tkinter, Geospatial Data Abstraction Library (GDAL), and NumPy. Tkinter is the standard GUI development package and acts as a Python interface to Tk/Tcl GUI toolkit. The Tkinter library is used to construct the three GUI display windows, menus, labels, and other user interface objects. The NumPy and GDAL packages are used to create the necessary data objects and functions attached to each GUI window. GDAL is used for raster and vector data translation and processing. NumPy is designed for Python scientific computing and allows for the construction and processing of N-dimensional array object (matrices). The following subsections describe the construction and algorithm of the each GUI window and its associated data objects and functions.

4.4.2 Overview Window: The Input Data

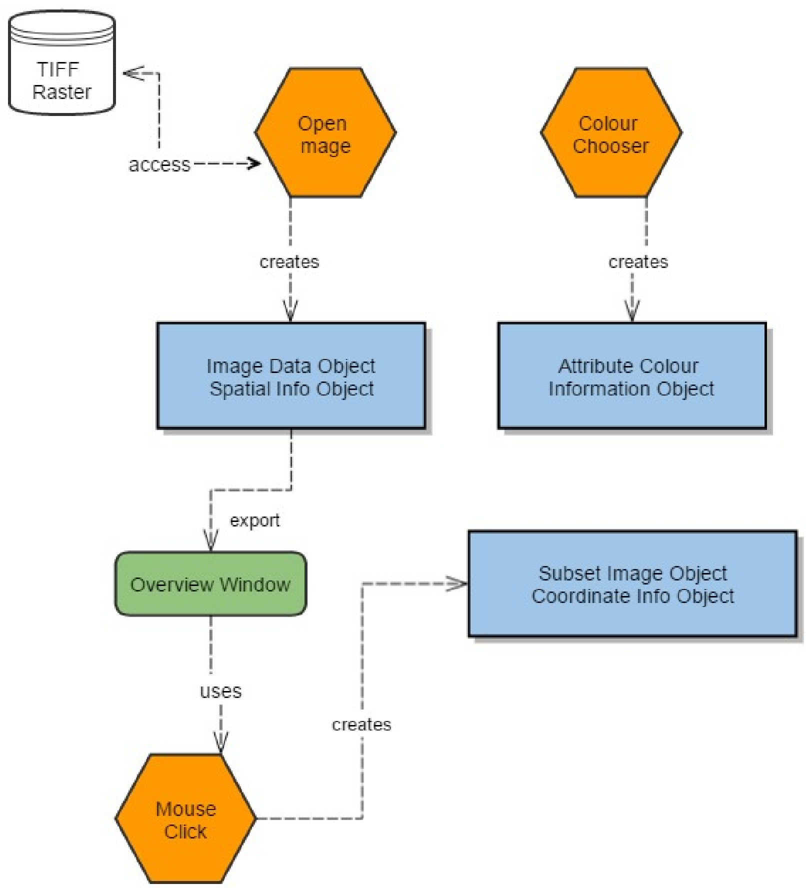

Figure 4.3: The Overview Window components focus on handling input TIFF images.

The Overview Window in Figure 3.3 and its associated components focus on the opening, retrieval, and processing of input data for other parts of GeoIcon Viewer (Figure 4.3). There are two input data that are required to create the icon image map and image layers chart: the TIFF image and the attribute selections information. The GeoIcon Viewer has a specific handler, highlighted in orange in Figure 4.3, for each input data object. The Open Image handler is for input image files and the Colour Chooser is for the attribute selection preferences. The Mouse Click is a Selection handler embedded in the Overview Window and allows users to pick out regions of interest. The following sub-subsections describe the implementation of the input data handlers and the Overview Window itself.

Input Data Handler: Open Image

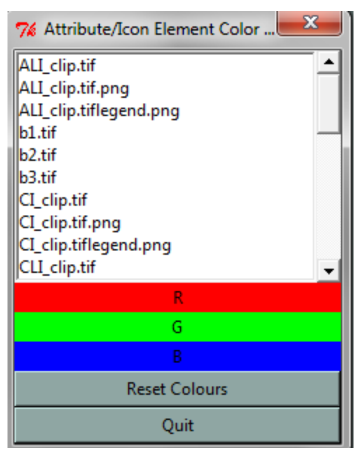

Figure 4.4: The dialog window for creating a false colour composite.

The Open Image is a handler for input image files and is implemented as the three sub-menu options discussed in Section 3.3. The Open Grayscale and Colour Image options bring up a File Browser window where users navigate to and select the appropriate input TIFF image. The third option, Creating Colour Composite, brings up a dialog window where users select the spectral band for each colour channel as shown in

Figure 4.4. The address of the grayscale or colour image is returned for the first two options. The Colour Composite function has a slightly more complex procedure. The matrix of each image selected is retrieved and used to fill the three dimensional matrix of the new composite image. The result of all three methods is an address to the appropriate image on the storage device even though each Open Image option has a different algorithm.

As stated in the System Design chapter, the program must be able to handle TIFF images as input data. The Open Image handler is designed specifically for that purpose. Implemented using the Geospatial Data Abstraction Library (GDAL), the Open Image handler reads, accesses, and retrieves data from the selected image via the sample code shown below. These steps return the spatial projection, georeference coefficients, and the image matrix. These objects are formatted to the appropriate data structure and passed to the Overview Window itself.

```
>>> dataset = gdal.Open('OverviewImage.tif', gdalconst.GA_ReadOnly)
>>> projectionInformation = dataset.GetProjectionRef()
>>> geoTransform = dataset.GetGeoTransform()
>>> geoTransform
(-81.4870785286949, 8.245251919505293e-06, 0.0,
 42.926336051546734, 0.0, -6.059970599398229e-06)
>>> raster = dataset.GetRasterBand(1)
>>> rasterData = raster.ReadAsArray(0,0, raster.XSize, raster.YSize)
```

Spatial Projection, Georeference Coefficients, and Image Matrix

The spatial projection is standardized using the Well Known Text(WKT) format and contains information about the coordinate system and the datum used for the image.

The projection information is retrieved and stored during program execution in a String object which is a sequence of characters. There are two coordinate systems: geographic and projected. The projection information and georeference coefficients are required for coordinate transformation.

The georeference or affine transformation coefficients are used in coordinate transformations between raster image space (pixel/line) and the georeferenced coordinate space using Equation (4.1) shown below. In the example shown above, the first (-81.4870785286949) and fourth (42.926336051546734) coefficient values are the longitude and latitude coordinate of the top left image pixel. The second (8.245251919505293e-06) and sixth (-6.059970599398229e-06) coefficient values are the horizontal and vertical pixel resolution. Lastly, the third (0.0) and fifth (0.0) coefficient values represent the rotation angle value.

$$X_o = a_1 X_s + b_1 Y_s + C \tag{4.1a}$$

$$Y_o = a_2 X_s + b_2 Y_s + C \tag{4.1b}$$

The last object returned by the Open Image handler is the image matrix. The matrix object, an instance of the NumPy.ndarray class, is constructed using a data structure called List. Each row of the image matrix corresponds to a list stored within one main list. The number of lists in the main list is the height of the image matrix. For example, if an image has the dimension of 512 by 512 pixels, the main list (image matrix) contains 512 sub-lists. Within each of the 512 sub-lists, there are 512 elements. The data type of the matrix elements vary from integer to complex numbers depending on the image. For a gray-scale image, a two dimensional matrix is returned and its shape is represented by a tuple of (1, YSize, XSize), where the numbers represent dimension, height, and width respectively. For a coloured image, the matrix shape is (3, YSize, XSize).

Overview Window

The Overview Window is one of the three GUI windows in the GeoIcon Viewer (Figure 3.3), and is an instance object of the CanvasFrame class which is derived from the Tkinter.Frame class as shown in Figure 4.5. A Tkinter.Frame class object is a rectangular interface region on the display screen, and serves as a container for other user interface components or widgets. For a CanvasFrame class object, the only widget contained within is a Tkinter.Canvas object, which provides a range of graphic capabilities including drawing and displaying graphs, geometric objects, and images. As a derived class of Tkinter.Frame, the CanvasFrame class inherited the class methods of Tkinter.Frame. The CanvasFrame class also contains its own class methods as shown in Figure 4.5.

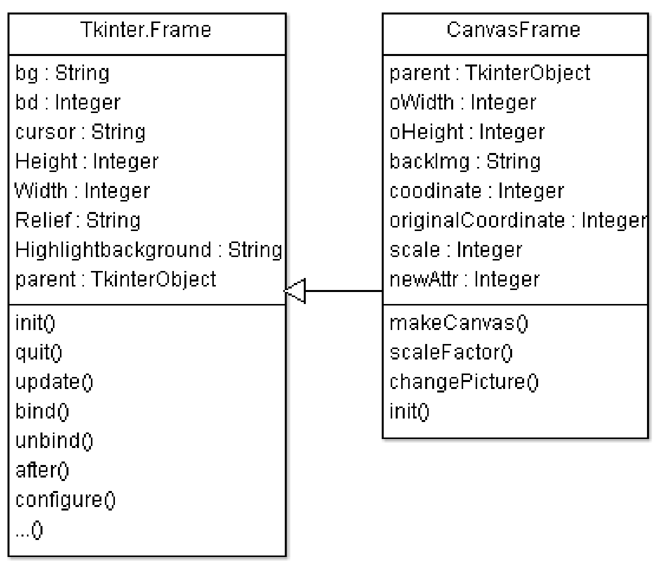

Figure 4.5: Overview Window is an instance of the CanvasFrame class, which is derived from the Tkinter.Frame class.

The Overview Window (OW) is designed to display the input TIFF image and allows for the selection of regions of interest. The zoom feature is dependent on this selection process since the region of interest is magnified and displayed as an icon image map in the Visualization Window. The OW receives the three data objects discussed in the previous sub-subsection: spatial projection, georeference coefficients, and the image matrix. The method changePicture() uses the matrix object to display the image. The image is resized before being displayed if the matrix height or width differs from Overview Window's dimensions. The scaling coefficient for both X and Y direction are calculated via Equation (4.2), and are used during coordinate transformations. As stated in the System Design chapter, an image may appear to be too dark without any enhancements depending on its data distribution. Hence histogram equalization is applied before the image is displayed.

$$X_{scale} = float(OW_Width)/float(Mwidth) \tag{4.2a}$$

$$Y_{scale} = float(OW_Height)/float(Mheight) \tag{4.2b}$$

There are three coordinate systems used within the GeoIcon Viewer: spatial, image, and the resized image. Conversions between the different coordinate systems are vital to the selection and spatial data query process. When users select a specific location via mouse click, the coordinate of the click event is defined within the resized image coordinate system. The scaling coefficients are used to convert the coordinates between the resized and the actual image. There is a loss of locational precision due to the rounding of calculation results. A single point on the resized image does not necessarily correspond to a single point on the original image, and vice versa. The actual coordinate is used to create a subset matrix of the original image. The geographic or projected coordinates of any selected locations are calculated using image coordinates, georeference coefficients, and the spatial projection. The result, Coordinate Info object, is passed on to the visualization processors along with the image subset matrix.

Colour Chooser

The Colour Chooser handler is designed for user customization of icon element colour for a more intuitive representation and returns the Attribute Colour Information (ACI) object. Shown in Figure 3.7, the handler, implemented as a dialog window, stores the selected attribute name and the corresponding colour within a data structure called Dictionary, where each attribute name is used as an index key. Each key references a tuple containing RGB values. The size of the dictionary is the number of attributes selected, which in turn determines the size of each icon.

4.4.3 Visualization Processors

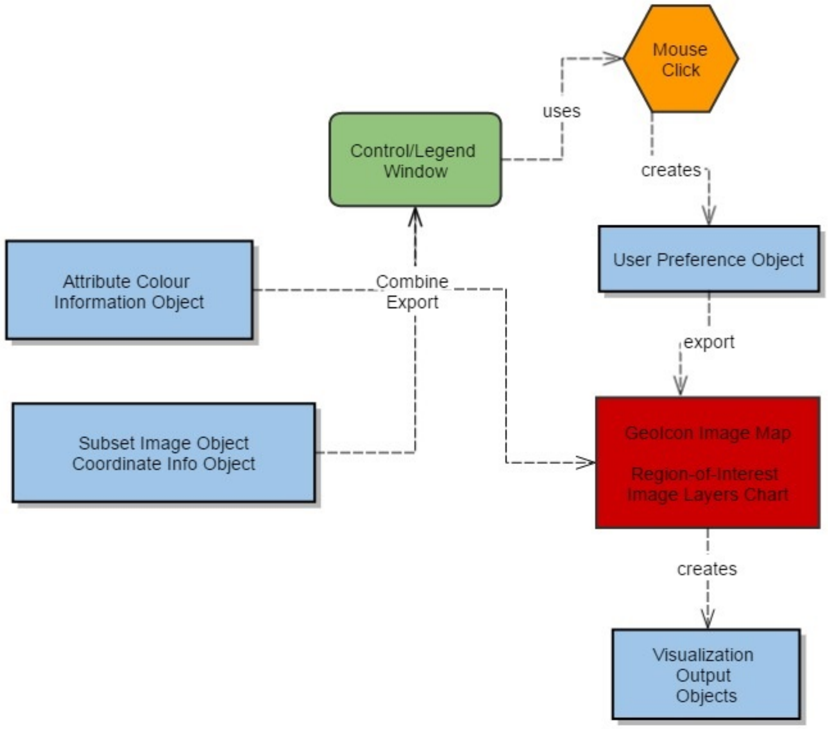

Figure 4.6: All the components and methods required to transform raw data into an icon image map and an image layers chart.

The Overview Window and its associated components generate three different data objects. The Attribute Colour Information object is a dictionary containing selected attributes and their corresponding colour. The Subset Image object is the matrix of an image subset that is created by the mouse click event in Overview Window. Lastly, the Coordinate Info object is a tuple containing the spatial coordinates. These data objects are passed on to the Visual Technique Processors (highlighted red in Figure 4.6). This component, as its name implies, creates the two visualization output: icon image map and image layers chart.

Control/Legend Window

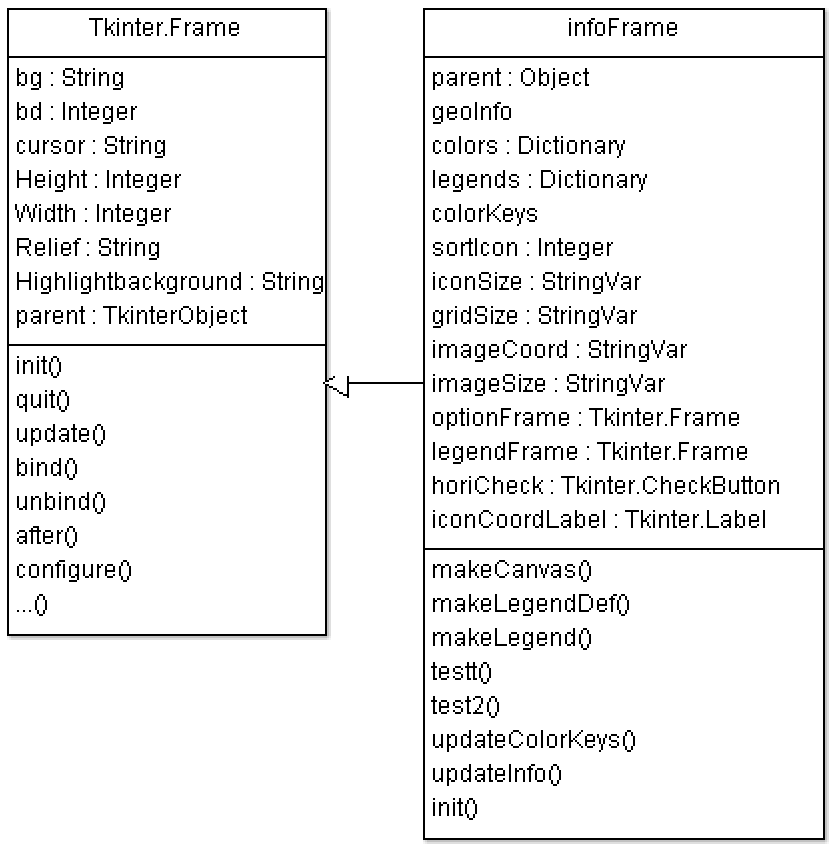

Figure 4.7: The Control/Legend Window, much like the Overview Window, is an instance object of a class (infoFrame) derived from Tkinter.Frame class.

The Control/Legend Window (CLW) in Figure 3.3 is a member of the infoFrame class which is derived from the Tkinter.Frame class as shown in Figure 4.7. The infoFrame class is implemented as a customized Tkinter.Frame object that contains two standard Tkinter.Frame objects. The first sub-frame contains a number of Tkinter.Label and Tkinter.Text objects that display information about the visualization output. The other sub-frame is used to construct the legend and holds up to nine Tkinter.Label objects where the background colour and display text are changed accordingly.

The purpose of the Control/Legend Window (CLW) is to support the visualization results and aid users in visual exploratory analysis by displaying the spatial coordinate and an attribute legend. The information for each legend element is queried from the Attribute Colour Information (ACI) dictionary. In addition, the Control/Legend Window allows users the option to change the orientation of icon elements. The User Preference Object, which is an integer, is passed to the GeoIcon Image Map visualization processor when the orientation option is selected.

GeoIcon Image Map

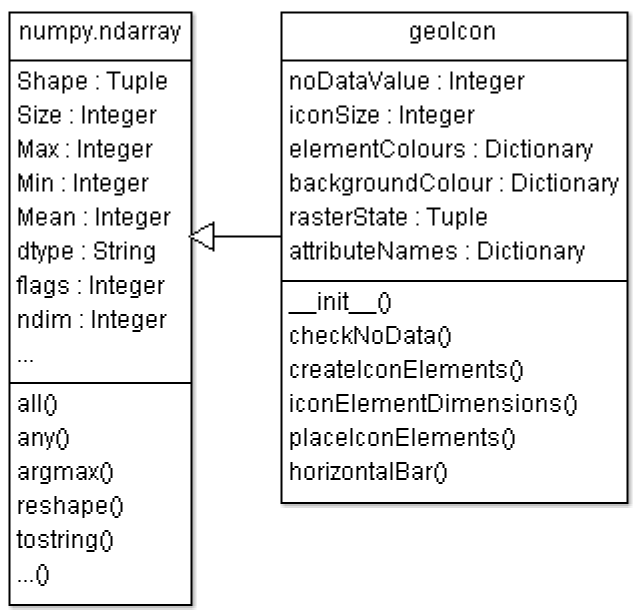

Figure 4.8: The GeoIcon class is derived from Numpy.ndarray class.

The GeoIcon class is constructed based on the icon design shown in Figure 3.6b-d. The icons and the icon image map are both implemented as matrices for display as an image. The size of each icon matrix is dependent on the number of attributes selected and the display screen size. The dimensions of the icon image map equals the dimensions of the input image subset multiplied by the icon size. The GeoIcon class is derived from the NumPy.ndarray class as shown in Figure 4.8. Each icon object is created by calling the GeoIcon class constructor method init(), which receives several input data objects containing details about attribute names, icon element colours, raw data values, and the background colour. The class methods query the input data objects to construct and place each icon element within the icon matrix.

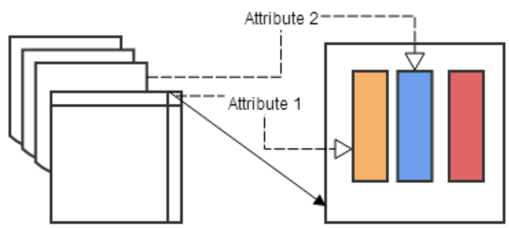

Figure 4.9: The transformation of multivariate attribute data into icon elements.

All elements in the initial icon matrix are set to zero. The iconElementDimensions() method uses the icon size and the number of attributes to calculate the pixel dimensions of all icon elements. A class method called createIconElements() transforms attribute values into icon elements based on the algorithm shown in Figure 4.9. This class method queries, returns, and standardizes the required attribute values using Equation (4.3). Each standardized value is multiplied by the pixel area of the icon element to determine the fill amount. The icon matrix is three dimensional and corresponds to RGB channels. For each dimension, the icon elements are filled with the corresponding RGB colour values selected by the user. The placeIconElements() method calculates the position of each icon element and places it accordingly. The unfilled pixels are set to the value of the corresponding raster cell from the unaltered overview image.

$$X'_{ij} = \frac{X_{ij}}{X_j^{max}} \tag{4.3}$$

Similar to an individual icon, the icon image map is a three dimensional matrix where a specific block is modified every time a new icon is created. The modified section in the icon image map is determined by the raster location within the input data matrix. The icon image map matrix is passed to the Visualization Window for display once an icon has been created for every raster in the input data.

Region-of-Interest Image Layers Chart

The Region-of-Interest Image Layers Chart method is designed to support the GeoIcon Image Map visualization method. Specifically, it is designed to address the issue of icon element size differentiation when the standardized values are very similar. It is reasonable to expect data values in a neighbourhood to be similar based on Tobler's First Law of Geography (Tobler, 1970). The Region-of-Interest Image Layers Chart method uses colour rather than size to represent quantity differences. Each attribute is visualized as an image layer bar based on the design shown in Figure 3.8. The pixels are ordered by their longitude and latitude within each image layer bar. Once mapped, the colour of each pixel illustrates the attribute value. Standardizing input values using the statistics of the input subset, rather than of the whole image, allows for greater colour distinctions between input values. The colour ramp of each image layer bar ranges from the icon element colour for the maximum value, to white for the minimum value. The rate of change in colour between pixels is dependent on the data range of the input matrix.

There is an issue with image layer bars created using the above algorithm where the size of each image layer bar is too small for visual examination or data query. The size of each image layer bar is the size of the input matrix using a one to one mapping logic which is relatively small compared to the display resolution. Each input matrix element or raster cell is represented by one pixel, and thus is hard to distinguish from others. In addition, user queries via mouse clicks become a tedious and error-prone task as users have to click precisely on the correct pixel. The size of each image layer bar is dependent on the number of attributes visualized. Each raster is magnified eight times in both the X and Y direction for six or more attributes, and ten times for less than six attributes. This magnification allows users to select and query the correct raster cell with ease, and also allows for a more visually effective image.

Region-of-Interest Image Layers Chart Display

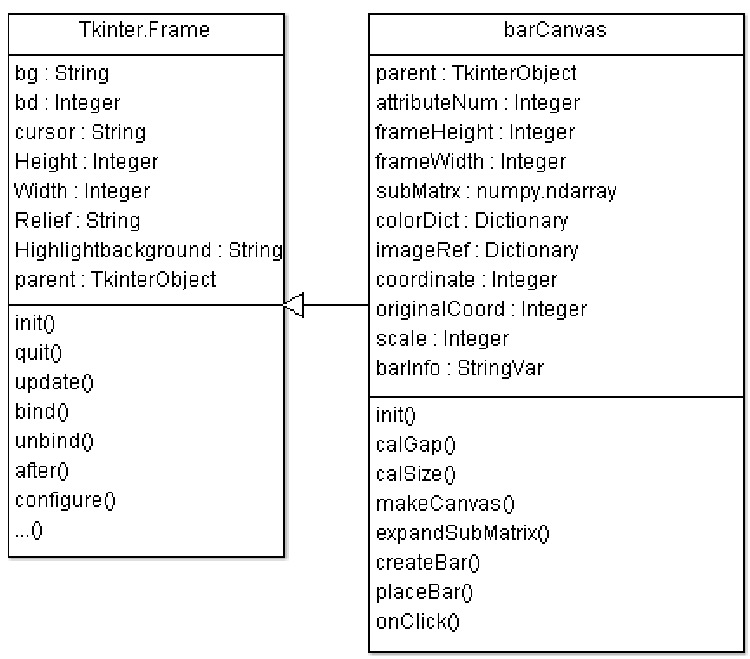

Figure 4.10: The barCanvas class is derived from Tkinter.Frame class. The image layers chart is created and displayed when the barCanvas object is constructed.

The sub-subsection above described the algorithm for the implementation of the Region-of-Interest Image Layers Chart method, and this section describes the actual implementation. Each image layer bar is not a distinct data object of any Python class. The Region-of-Interest Image Layers Chart method is implemented as a derived class of Tkinter.Frame called barCanvas as shown in Figure 4.10. Much like the CanvasFrame class, the barCanvas class uses a canvas to display the output. The same image subsets used to create the icon image map are used to construct the image layer bars. The icon image map is a complete visualization output object, but the image layer bars require additional graph elements such as X and Y axis, labels, titles, and legends to form the image layers chart. The image layer bars are merged with the display window to form the barCanvas class in order to make the code contained and easier to manage.

The creation of the each image layer bar follows the algorithm described in the previous sub-subsection. The placeBar() method calculates the location for each bar taking into account the number and size of the image layer bars, and places each bar accordingly on the canvas. A X and Y axis is drawn in the display canvas, along with a colour ramp legend for each image layer bar. The barCanvas class has a user event handler for data query called onClick(). Unlike the icon image map, there are other data objects including lines and text in the display space. The pseudo matrix query method that was implemented for the icon image map, explained in the next subsection, cannot be used for the image layers chart. The onClick() method uses the position of each image layer bar as a reference key. The handler first evaluates whether the user has clicked within a image layer bar. If true, the event coordinate is used to determine the reference key value and the appropriate input data value is returned.

4.4.4 Display of Visualization Outputs

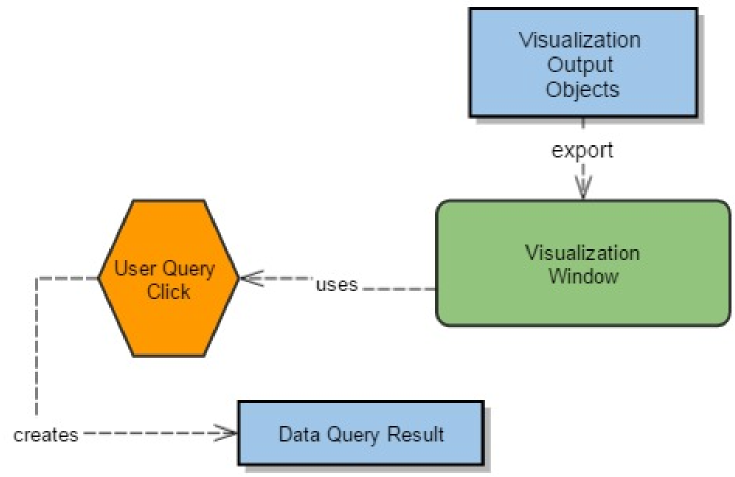

Figure 4.11: VW is the third GUI window and is designed to display outputs and provide dynamic user querying.

The Visualization Window is the third and final GUI window (Figure 3.3) and is designed for displaying and querying the icon image map as shown in Figure 4.11. The icon image map is displayed in the Visualization Window, but the image layers chart is created and displayed within a dialog window created on user demand. The reason for this setup is to provide a concurrent view of the study location with different visualization methods following the brushing and linking approach discussed in the System Design chapter. The Region-of-Interest Image Layers Chart method allows an easier differentiation between attribute values compared to the GeoIcon Image Map method. The VW, along with Overview Window, are used to implement the zoom feature. Region of interest selected in the Overview Window is magnified in size and detail within the Visualization Window.

Visualization Window

The Visualization Window is an instance object of the iconCanvas class. Similar to the other two GUI window classes, the iconCanvas class is derived from the Tkinter.Frame class with the difference being an extra parent class as shown in Figure 4.12. The iconCanvas class is constructed by further modifying the CanvasFrame class in order to minimize code redundancy.

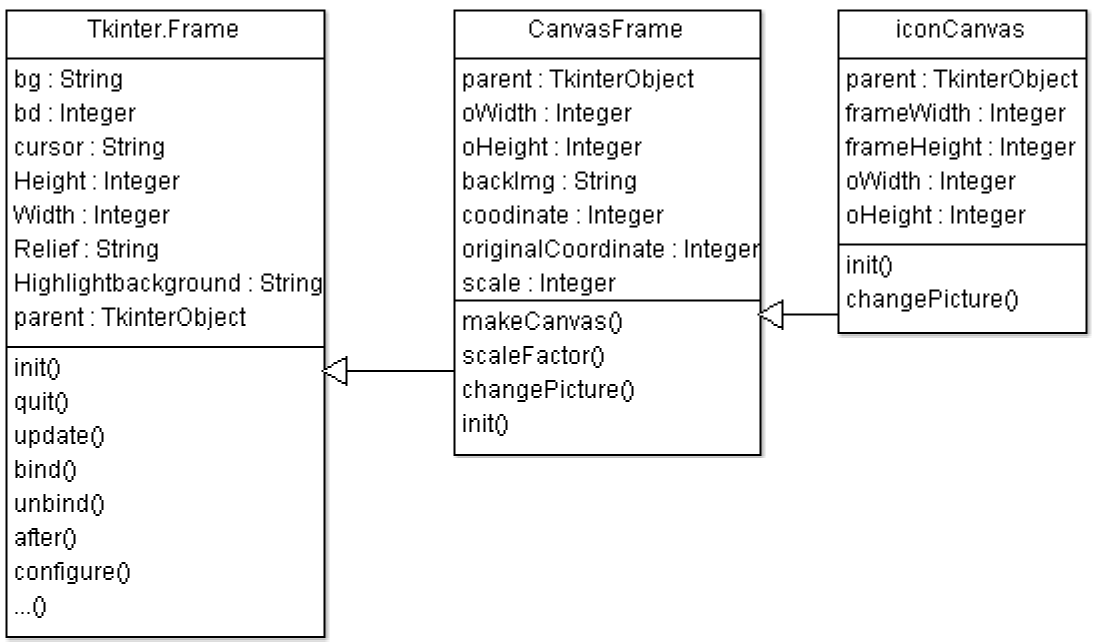

Figure 4.12: VW is a member of iconCanvas class, which is derived from the Canvas-Frame class.

The VW resizes the icon image map to fit the display space, and the scaling coefficients are calculated via the inherited method called scaleFactor(). The difference between the Overview and Visualization Window is in the interaction process. The dynamic data query system, one of the required program features, was designed around user interaction within the VW. The query system for the icon image map is implemented as a three dimensional matrix. A secondary pseudo icon is created alongside each real icon. Within the pseudo icon, each icon element dimension is filled with a reference key value, standardized input value, and the original input value respectively. The rest of the icon is then filled with a predefined null value in all three dimensions. The pseudo icons form a pseudo icon map that is the same size as the actual icon image map. When users click on the icon image map, the scaling coefficients are used to find the actual image coordinate which is then used to query the pseudo icon map. The query result is displayed as a tool-tip at the cursor location within VW.

4.5 Components Integration

There are three main GUI window components in the GeoIcon Viewer: Overview Window, Visualization Window, and Control/Legend Window. Each window component is derived from the Tkinter.Frame class. Once each GUI window and its associated functions have been implemented and tested, the next step is to integrate them into one complete graphic user interface as shown in Figure 3.3. In Tkinter, each GUI requires a root window, which acts the overall container for all GUI components such as frames, buttons, or labels. Integration of the three GUI windows is simply to pack them into a root window as one would do with standard Tkinter.Frame objects. This is allowed because all three GUI windows are derived from the Tkinter.Frame class and thus inherited all of its class methods. Linking up the data flow between each GUI window is simply the process shown in Figure 4.2, which describes how each GUI window and its associated handlers interacts with the rest of the program. There are many implemented program objects and processes that are not discussed in this chapter because the focus has been on the implementations of major feature requirements: the multi-window GUI, dynamic data query system, TIFF image processing, the new GeoIcon Image Map technique, and the Region-of-Interest Image

Layers Chart visualization method. The entire program is written in Python, and runs on Unix, Linux, and Windows. The next chapter demonstrates and evaluates the GeoIcon Viewer using a case study.

Chapter 5

Demonstration and Evaluation

The GeoIcon Viewer is a new visual exploratory program for multivariate spatial data. Based on the IconMapper System, the GeoIcon Viewer features an improved icon design (GeoIcon Image Map), a data query system, and the Region-of-Interest Image Layers Chart visualization method. The new icon design allows more variables to be visualized simultaneously as well as aiming to remove the unwanted visual effects shown in Figure 2.7. The data query system is linked with the icon image map output and returns information about the selected icon element. The Region-of-Interest Image Layers Chart approach is designed to complement the GeoIcon Image Map method. The three main program features were implemented as an integrated set of components using the Python programming language. Python was chosen because of its programming style, portability, and development efficiencies. This chapter focuses on the latter part in the fourth stage of the Waterfall model. A case method is used to demonstrate the GeoIcon Viewer's application using real world data, and allows for the evaluation of the GeoIcon Viewer.

5.1 Case Study and Data

The case study revolves around the study of a porphyry copper deposit through the detection of alteration minerals using remote sensing. The study area shown in Figure 5.1 is a mountainous region in Peru. The area of focus is bounded by the red polygon shown in Figure 5.1. This chapter does not address the geological science behind copper formations, but rather focuses on demonstrating and evaluating the GeoIcon Viewer.

Index Layer	Equation
Hydroxide (OH)	$\frac{band_2 * band_7}{band_6 * band_6}$
Kaolinite (KLI)	$\frac{band_4 * band_8}{band_5 * band_6}$
Alunite (ALI)	$\frac{band_7 * band_7}{band_5 * band_8}$
Calcite (CLI)	$\frac{band_6 * band_9}{band_8 * band_8}$
Quartz (QI)	$\frac{band_1 * band_{11}}{band_{10} * band_{12}}$
Carbonate (CI)	$\frac{band_{13}}{band_{14}}$
Stable Vegetation (StVI)	$\frac{band_1 * band_3}{band_2 * band_2}$
Mafic (MI)	$\frac{band_2}{band_{13}}$

Table 5.1: Mineral band ratio equations from Ninomiya et al. (2005) and Pour and Hashim (2012).

The input data is ASTER(Advanced Spaceborne Thermal and Reflection Radiometer). ASTER is a multi-spectral sensor that provides spectral data in fourteen different bands ranging from the visible to the thermal infrared portion of the electromagnetic spectrum. The ASTER data was packaged inside a .HDF file. Once extracted, the images were calibrated and ortho-rectified using a digital elevation model (DEM) and embedded ground control points, which is required due to terrain distortions caused by the lens distortion and sensor tilt. Alteration minerals associ-

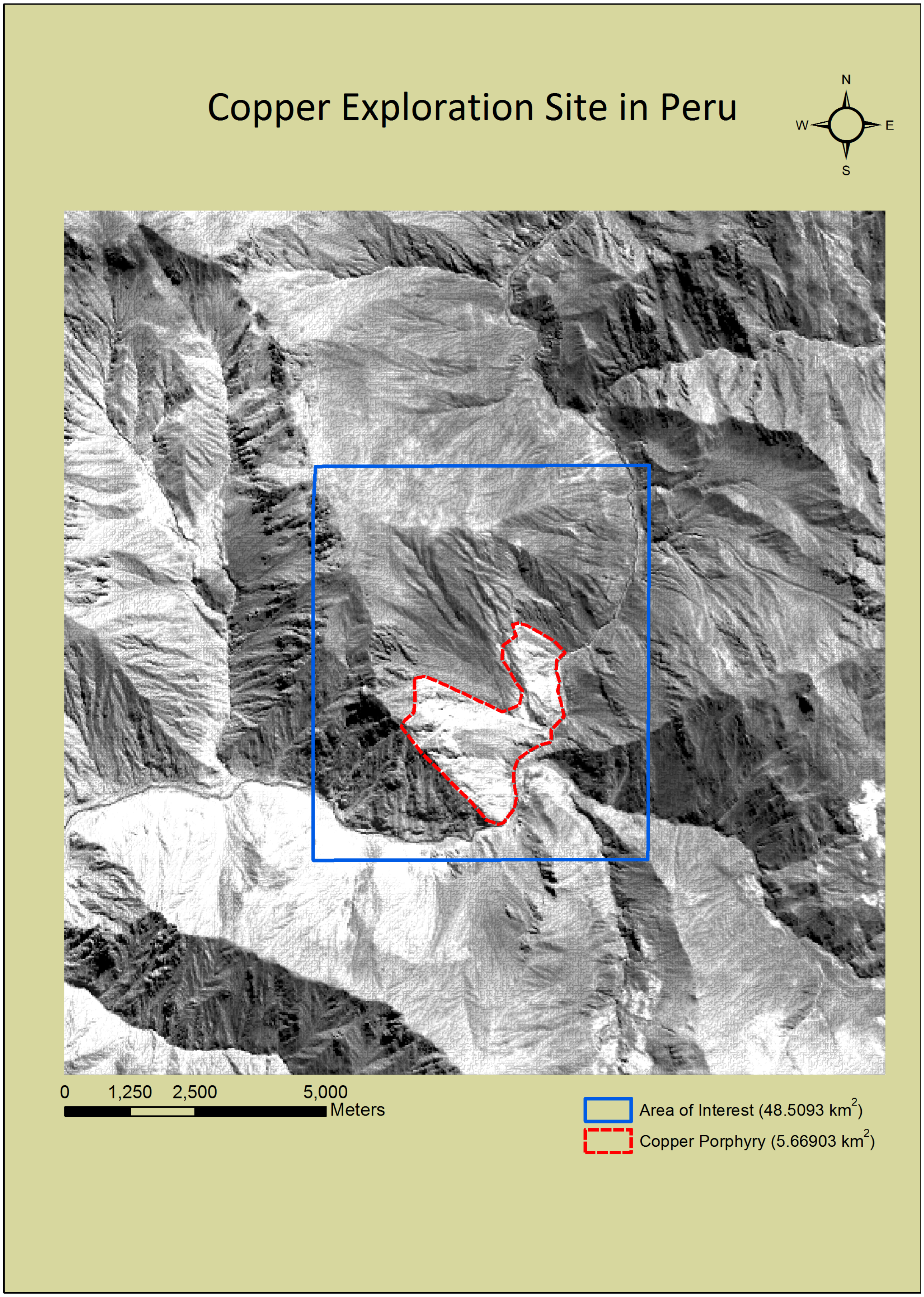

Figure 5.1: An image map of the case study area. The area contained in the blue rectangle is the area of interest (AOI) and the area in red is the main exposed portion of the copper porphyry.

ated with copper deposits are hydroxide(OH), kaolinite (KLI), alunite (ALI), calcite (CLI), quartz (QI), carbonate (CI), and mafic (MI).

An index or distribution image was created for each alteration mineral by using band ratio operations based on the mineral's spectral signature. Band ratio is a digital image processing technique that enhances contrast between features by emphasizing spectral differences. The band ratio equations shown in Table 5.1 were taken from Ninomiya et al. (2005) and Pour and Hashim (2012). They were used to calculate all the index images which were then saved as TIFF images. An index image for vegetation was created in addition to the seven alteration minerals. All of the seven index images contained values that can be correlated to the quantity of a particular mineral which allows for the ranking of mineral quantity at each location. While every raster cell has a value, only some rasters contain actual alteration minerals. The remaining raster cells contain values correlated to noise and non-mineral objects, and must be removed before the index image is of any use. The GeoIcon Viewer, like many other data processing or visualization software, is subject to the Garbage In and Garbage Out (GIGO) concept where erroneous input results in erroneous outputs. While the human visual and cognition systems are good at filtering out noise, there are limits. A thresholding and reclassification operation was applied to each index image and the results are presented below. Figures 5.2 to 5.9 provide a two and three dimensional perspective view of the distribution pattern for each attribute. It must be stated that the locations and the distribution pattern of each mineral are dependent on, and very sensitive to, the threshold value chosen, and thus the information presented in these figures are only the potential distribution patterns. In the following subsections, each data layer is examined and a determination is made as to whether the layer is suitable as an input for the GeoIcon Viewer.

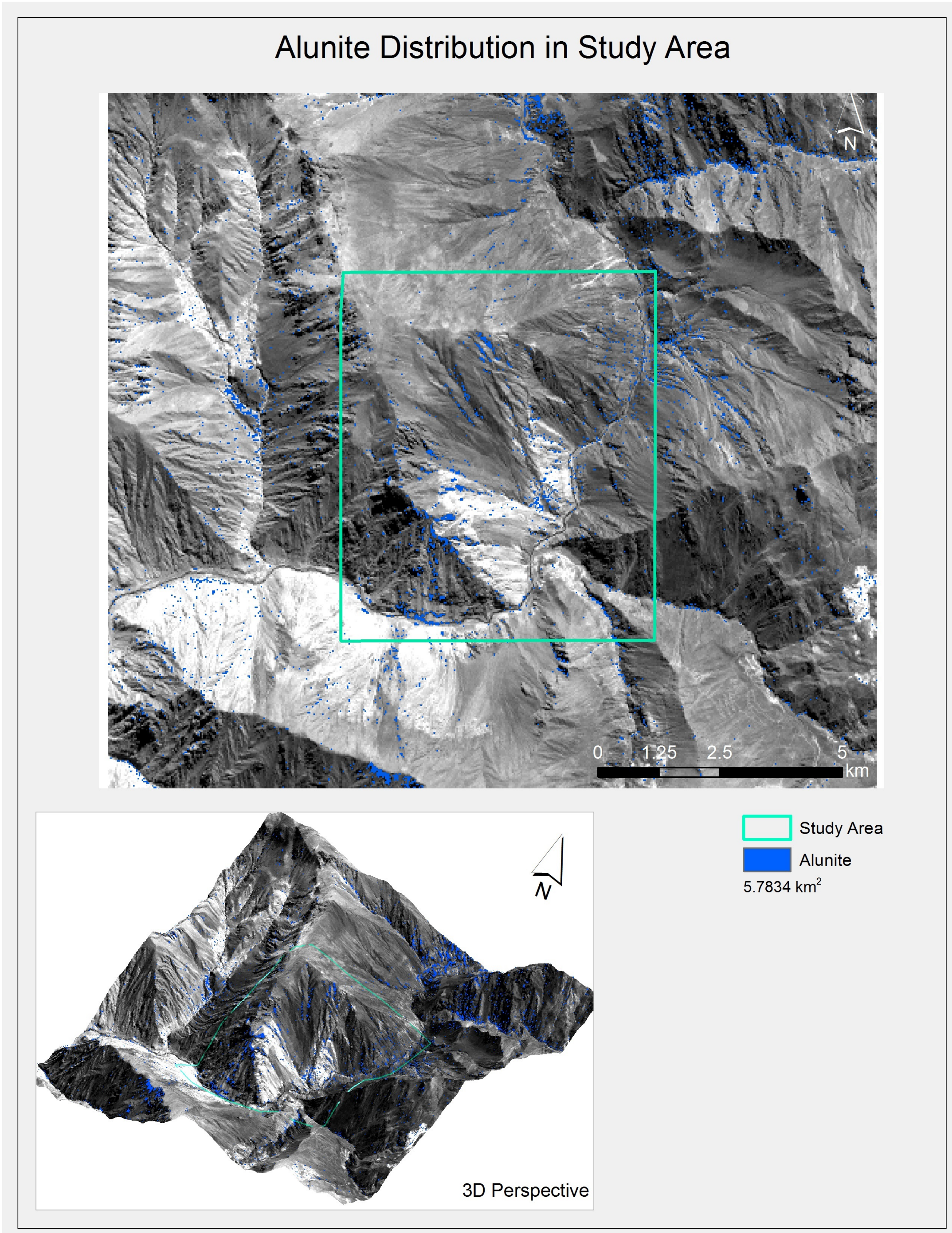

Figure 5.2: Potential distribution pattern of alunite within the study area.

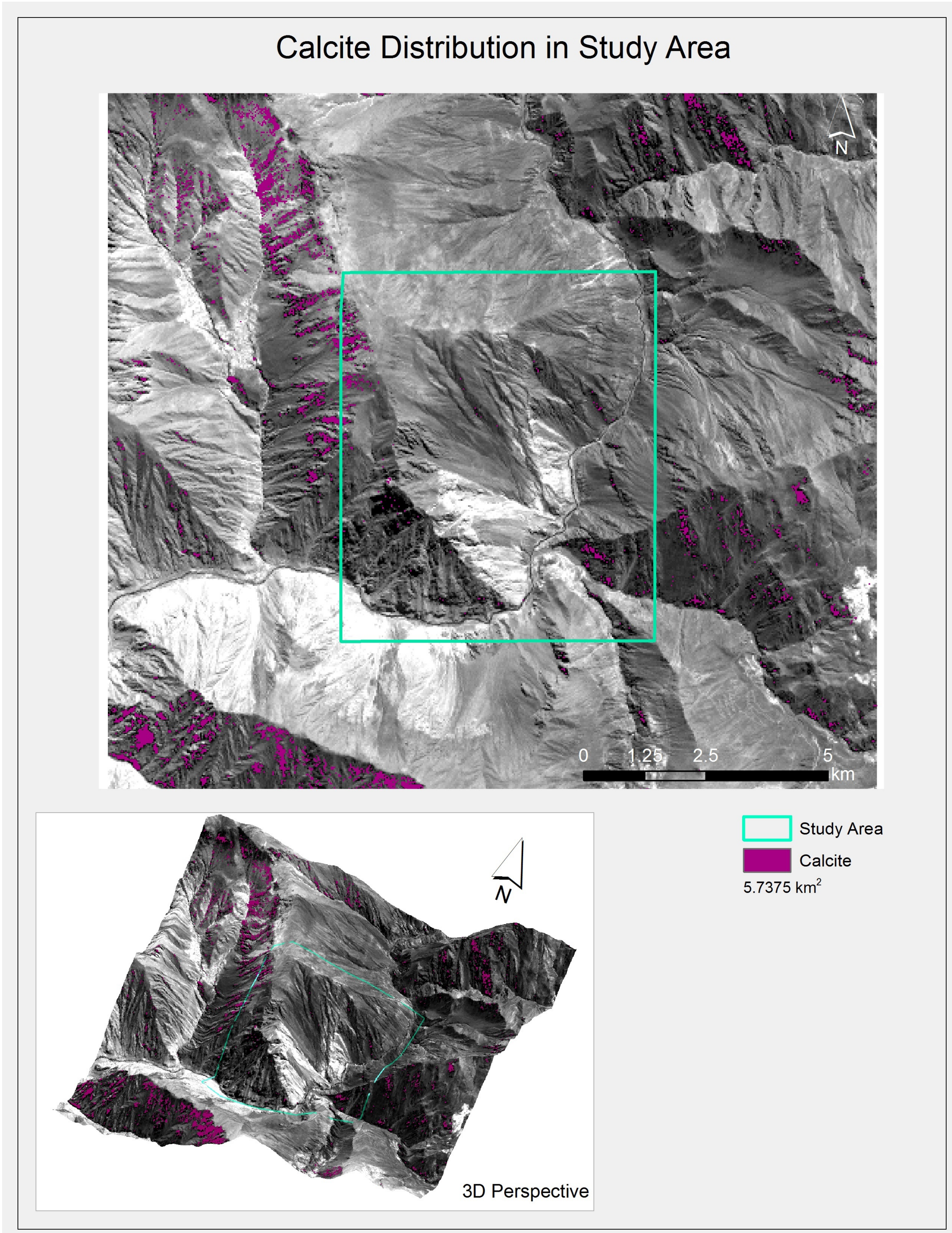

Figure 5.3: Potential distribution pattern of calcite within the study area.

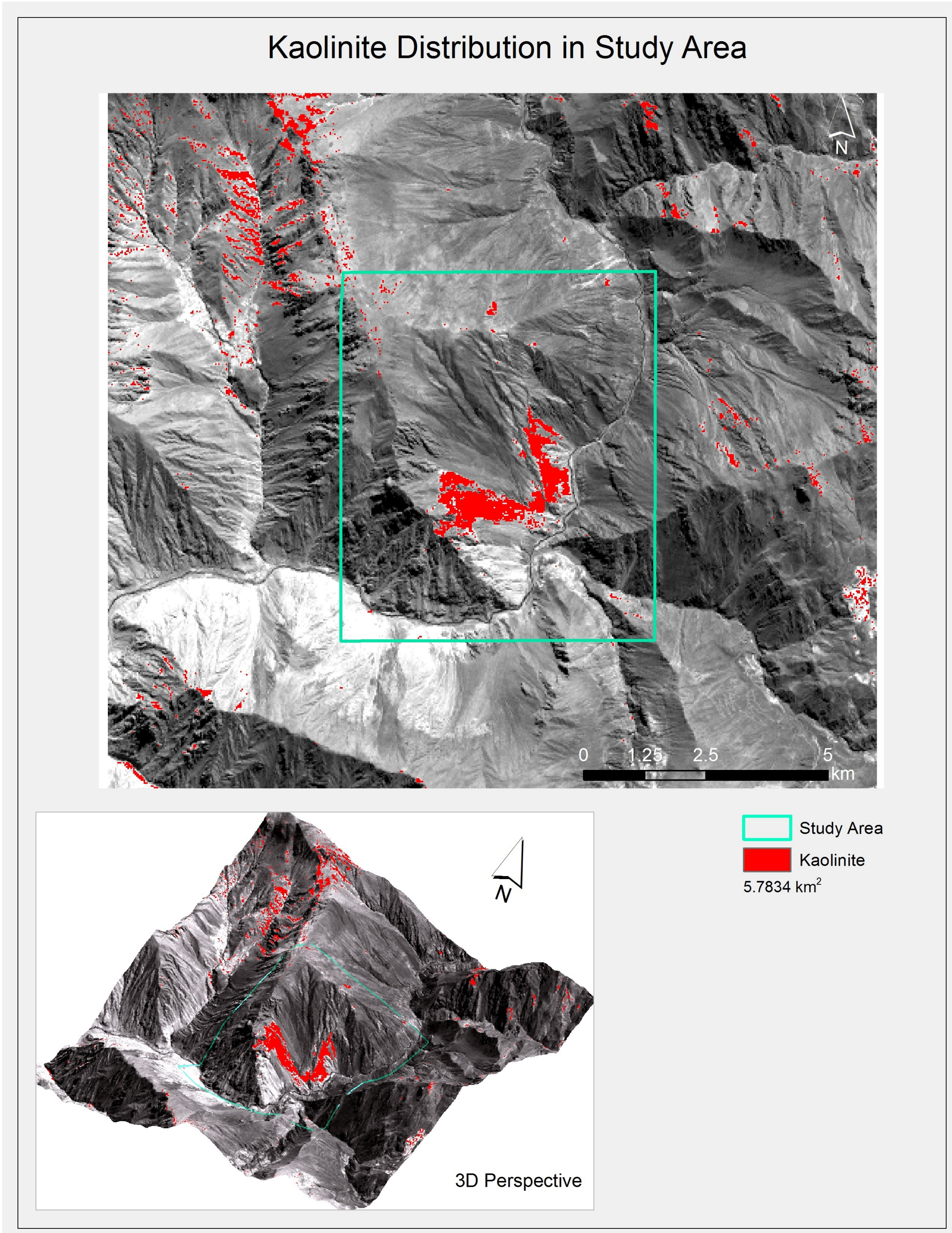

Figure 5.4: Potential distribution pattern of kaolinite within the study area.

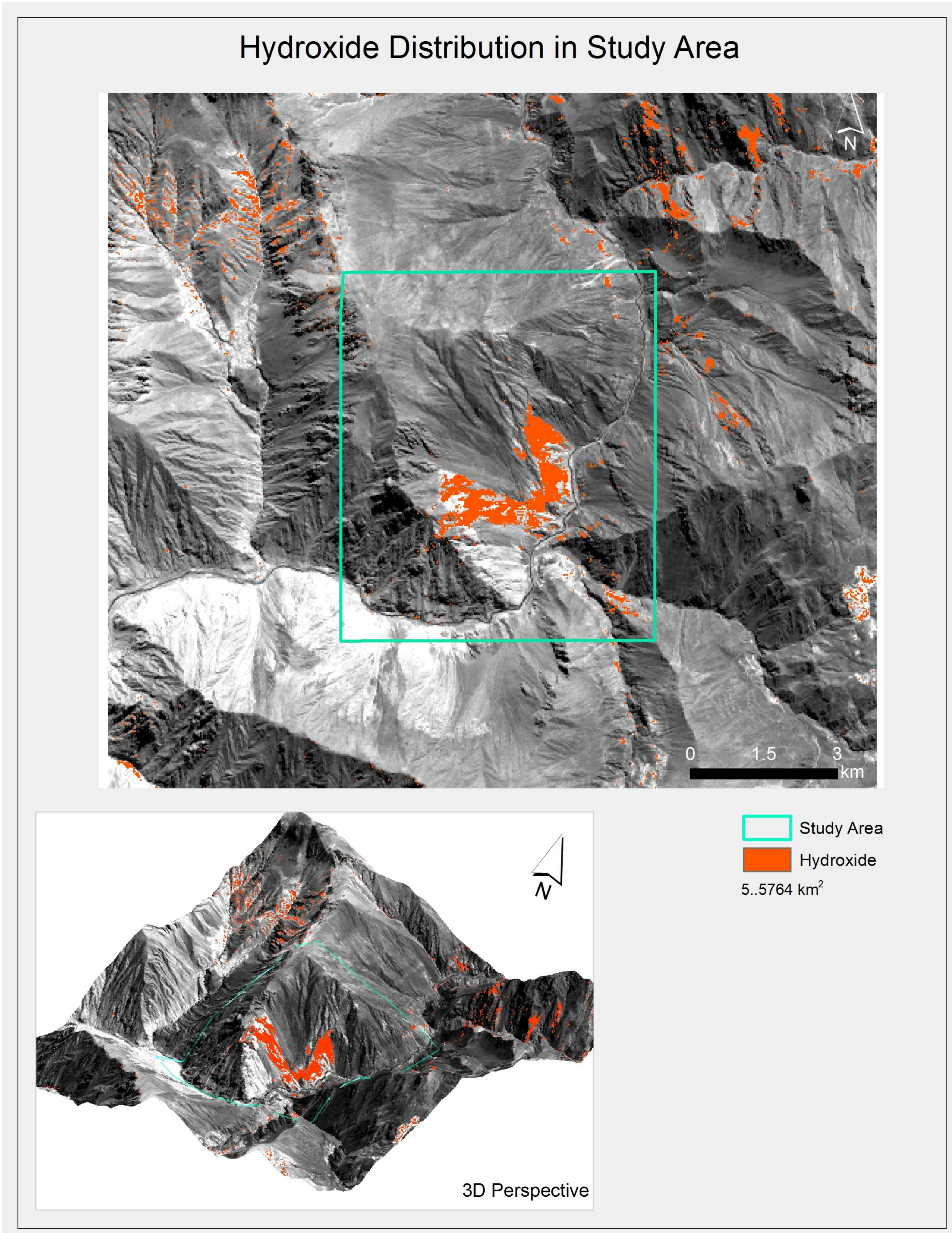

Figure 5.5: Potential distribution pattern of hydroxide within the study area.

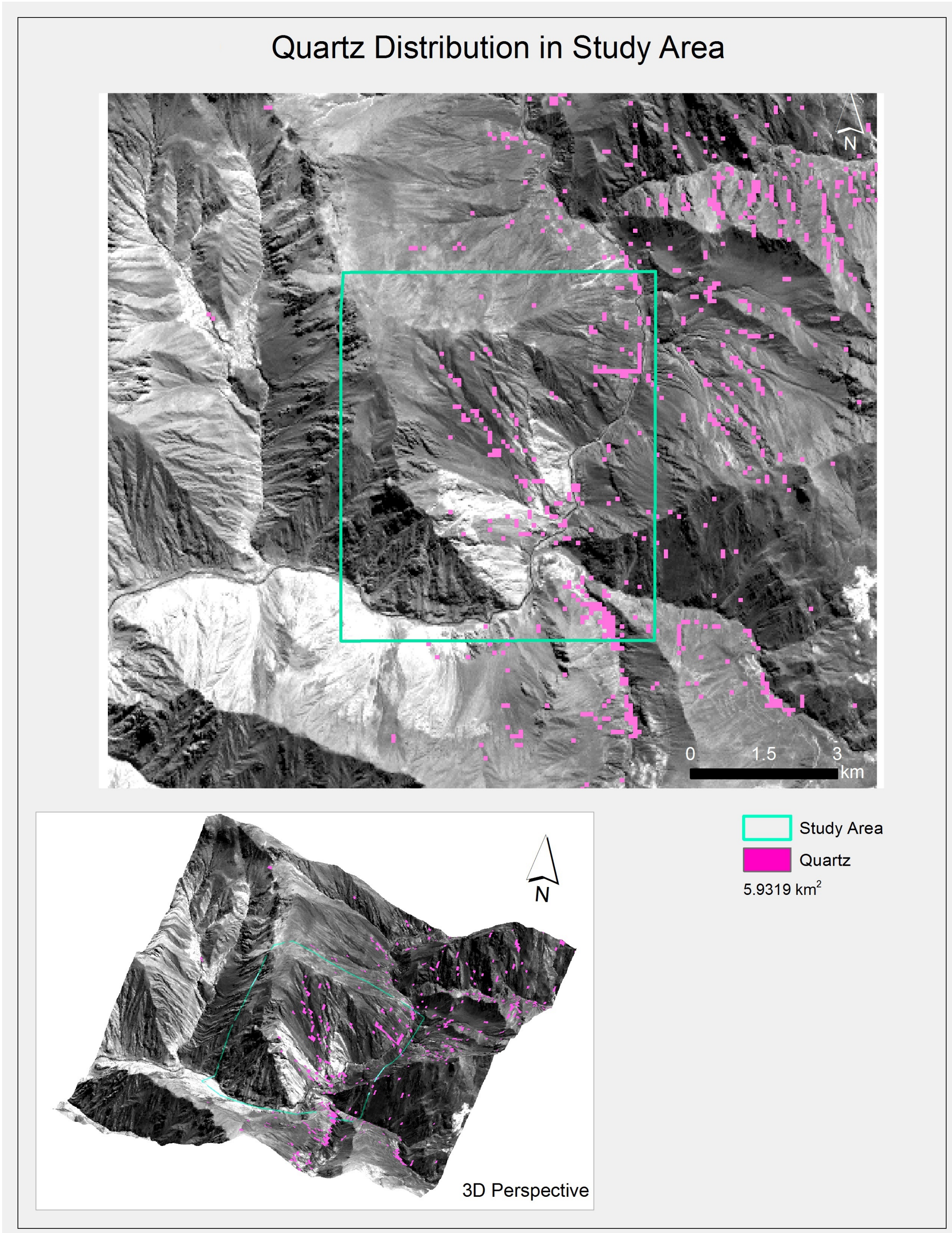

Figure 5.6: Potential distribution pattern of quartz within the study area. Notice the block effect due to the low resolution thermal bands.

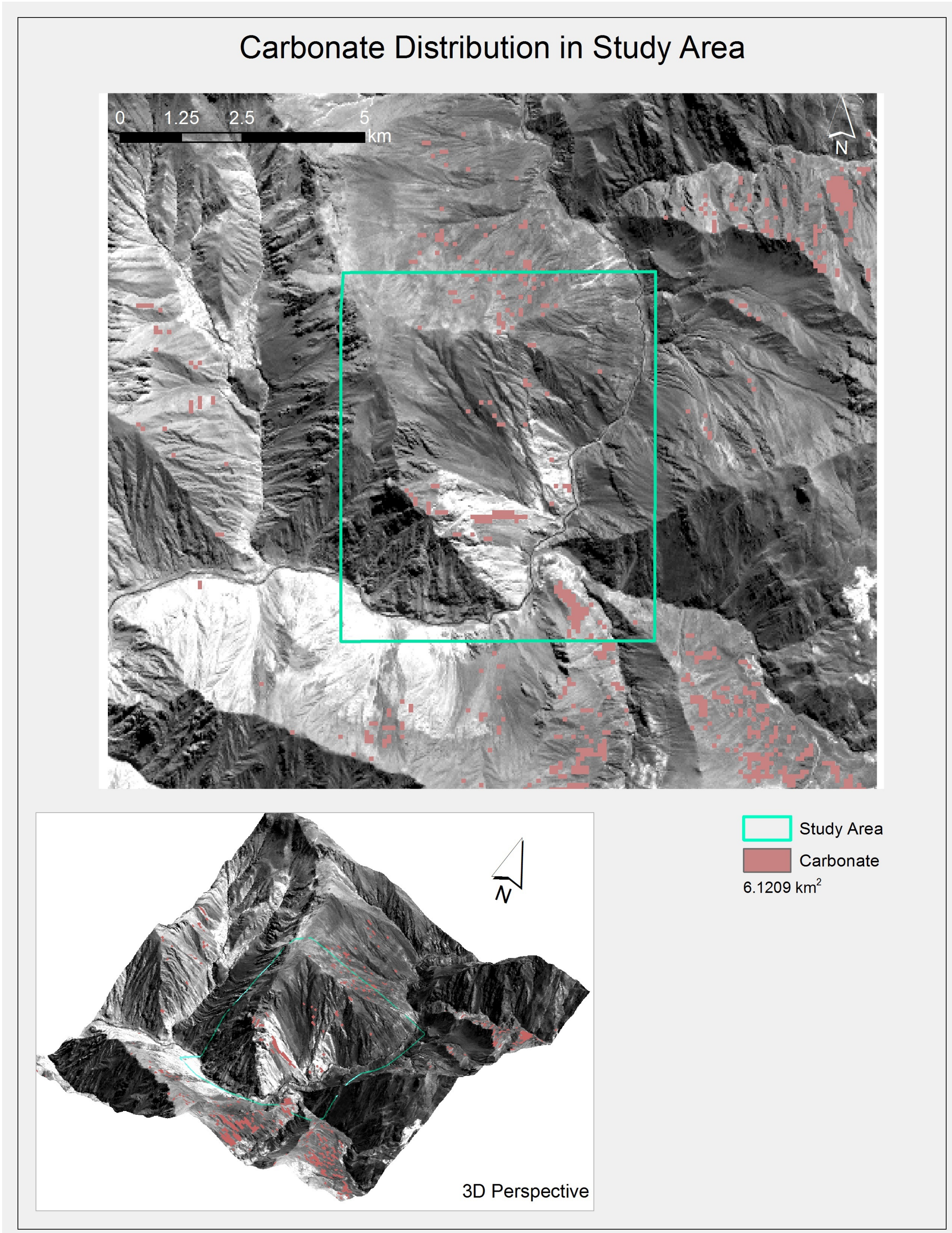

Figure 5.7: Potential distribution pattern of carbonate within the study area.

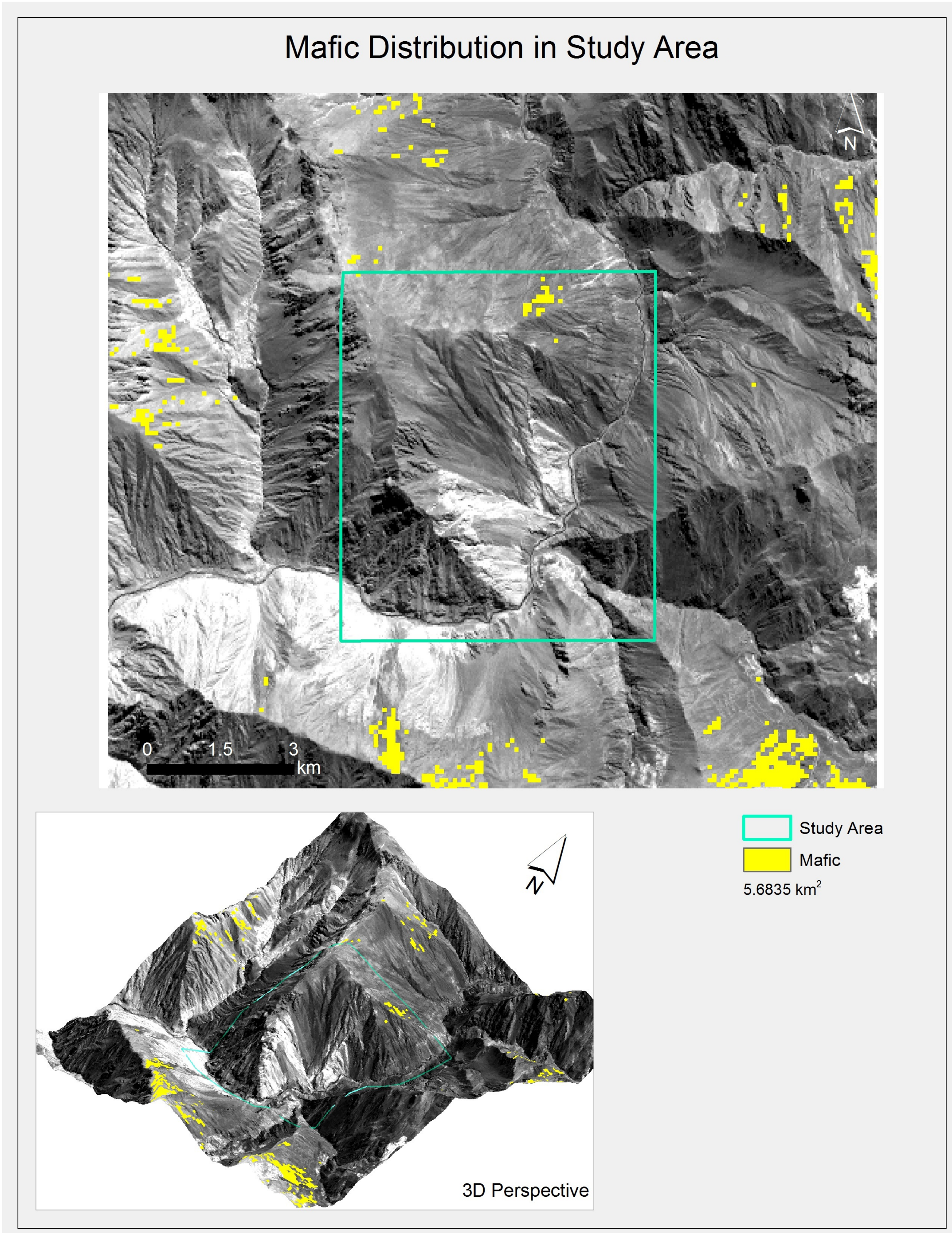

Figure 5.8: Potential distribution pattern of mafic within the study area.

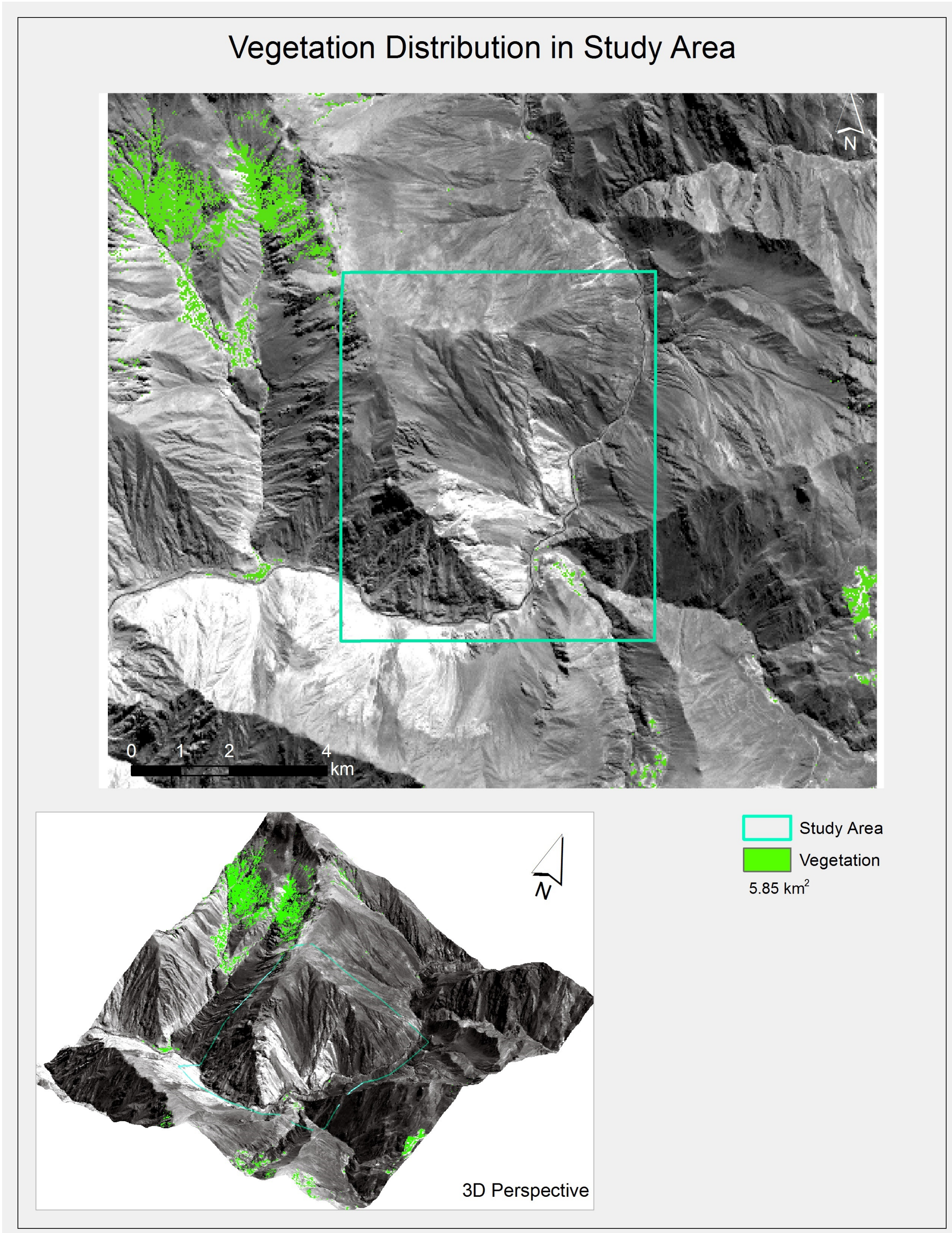

Figure 5.9: Potential distribution pattern of vegetation within the study area.

5.1.1 Alunite

Alunite, coloured blue in Figure 5.2, is distributed sparsely over the entirety of the study area. Looking at the two dimensional map, there seems to be a relatively higher amount of alunite in the northern and center region. Based on the 3D map, alunite is mostly located on the ridges and steep slopes, with small amounts near river banks. There is sufficient amount of alunite for the index image to be suitable as an input layer within the area of interest (AOI) and the copper porphyry. Similar to other regions of the study area, alunite within the AOI is found mainly on the steep slopes.

5.1.2 Calcite

The calcite distribution pattern is shown in Figure 5.3. Similar to the alunite distribution, the calcite is found on steep slopes. However, the distribution pattern of calcite is more clustered compared to the alunite. The main cluster is on the west side of the study area where the calcite is found in gullies and channels on the mountain slopes. Within the AOI, there is a limited quantity of calcite on the ridges and near the river bank. The calcite index image makes for a poor input layer due to the limited quantity within the AOI and the copper porphyry.

5.1.3 Kaolinite and Hydroxide

The distribution patterns of kaolinite and hydroxide, as shown in Figure 5.4 and 5.5, are very similar to each other but different from the other five minerals. There are three main clusters for these two minerals within the study area: northeast, northwest, and center. Based on the three dimensional maps, all three clusters are found on steep slopes with channels and gullies. Hydroxide seems to have a higher presence along river banks than the kaolinite (Figure 5.4 and 5.5). There are high amount of kaolinite and hydroxide in the AOI and the copper porphyry. Therefore these two layers are suitable as input data. Furthermore, both minerals seem to cover the entire copper porphyry area shown in Figure 5.1. The shape of the distribution pattern within the copper porphyry could be a result of geologic, tectonic, and geomorphic activities.

5.1.4 Quartz

The distribution pattern of quartz is shown in Figure 5.6. Only one side of the study area seems to contain quartz. The north to south boundary line of the distribution bisects the AOI. Similar to alunite, the majority of quartz is found on steep slopes with channels and gullies on river banks, which indicates potential mineral movements. Within the area of interest, there appear to be two linear distribution patterns. The main distribution starts from the northern ridge through the porphyry center and to the river bank. A secondary linear distribution of quartz starts from the southern ridge within the AOI, and cuts eastward through a portion of the porphyry towards the river. Since there are significant amounts of quartz within the AOI and the copper porphyry, quartz is a suitable input data layer for the GeoIcon Viewer.

5.1.5 Carbonate, Mafic, and Vegetation

The carbonate distribution pattern is shown in Figure 5.7. Similar to quartz, the majority of the carbonate is found on the eastern side of the study area and on steep slopes. There is also a linear distribution of carbonate originating from the southern ridge and progressing eastward or downslope towards the river. The carbonate index map is a suitable input layer as there is a significant amount of carbonate within the copper porphyry.

The mafic and vegetation are not suitable input layers based on the distribution patterns shown in Figure 5.8 and 5.9. Mafic is not found within or close to the copper porphyry. While there is vegetation that is close to the porphyry, vegetation itself is not usually directly relevant in mineral detection. The vegetation layer can be used as a mask to help eliminate potential noise in the mineral data layers. The vegetation layer is used as the eighth attribute to evaluate the implementation of the visualization methods in the GeoIcon Viewer.

5.1.6 Input Data for GeoIcon Viewer

In Figure 5.6, 5.7, and 5.8, the mafic, carbonate, and quartz pixel distributions appear to be blocky which is caused by the ASTER bands used to create the three index images. Only five attributes were deemed suitable as input based on the visual examination of the eight index images. The next section presents an evaluation of the program implementation, including the output visualization results: the icon image map and image layers chart.

5.2 Results and Analysis

As stated at the beginning of the chapter, the purpose of the case study is to demonstrate and evaluate the GeoIcon Viewer. For a systematic evaluation, the process followed the order of implementation and integration discussed in the previous chapter: Overview Window, Control/Legend Window, and lastly the Visualization Window. The design and feature requirements specified during the first two stages of the Waterfall model were used to evaluate whether the implementation and integration of each program component was successful. Using the GeoIcon Viewer to identify potential patterns in the input data also allows for the evaluation of the GeoIcon Image Map and Region-of-Interest Image Layers Chart visualization methods.

5.2.1 Overview Window

The purpose of the Overview Window and its associated components was to process and display TIFF input files. In addition, the OW allowed for the selection of regions of interest. Based on design requirements, the GeoIcon Viewer is constructed to open grayscale images as well as to create and open false colour composites as shown in Figure 5.10. Figure 5.10a contains a grayscale image of ASTER Band 3 (0.78-0.86 μm). In Figure 5.10b, three bands were selected to create a false colour composite of the study area. Both input images were 525 by 552 pixels, and were resized to 500 by 800 pixels to fit the Overview Window.

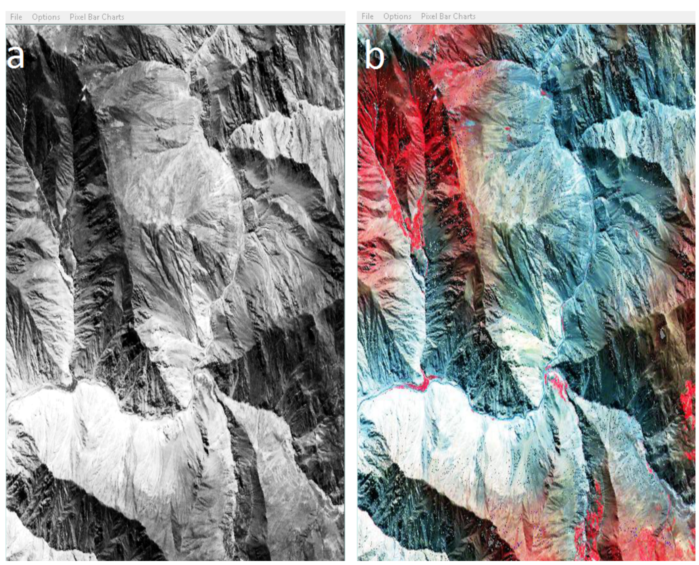

Figure 5.10: Overview Window image: (a) A grayscale image of ASTER Band 3 with histogram equalization. (b) A false colour composite created using ASTER band 1,2, and 3.

5.2.2 Colour Chooser and Control/Legend Window

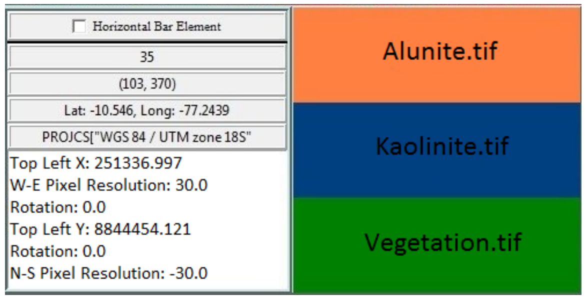

Figure 5.11: The chosen attributes, icon element colours, and location coordinates are shown in the Control/Legend Window.

Once the input image was selected, the next step was to choose the attributes and their corresponding icon element colour. The GeoIcon Viewer supports the visualization of up to nine attributes simultaneously. The information about the attributes and their icon element colour was used to create a legend and is a required input for the visualization processors. The coordinate query feature, which returned the geographic or projected location of the user focus, as shown in Figure 5.11, is enabled once the attribute selection process has been completed.

5.2.3 Visualization Window: Icon Design

The GeoIcon Viewer featured a new icon design where the icon size changed depending on the number of attributes visualized. While the IconMapper System only allowed for five attributes to be visualized simultaneously, the GeoIcon Viewer allows for up to nine attributes to be visualized within each icon. Three icon size configurations were implemented to accommodate the range of icon element numbers. Figure 5.12 shows the three icon sizes with two different icon element configurations. Within each icon, pixels outside the icon elements were set to a specific background colour value.

The size and position of the icon elements were automatically configured which leads to an effective use of the display icon pixel space. The end goal of the new icon design is to be able to visualize more attributes simultaneously and to reduce the unwanted visual effects that were shown in Figure 2.7.

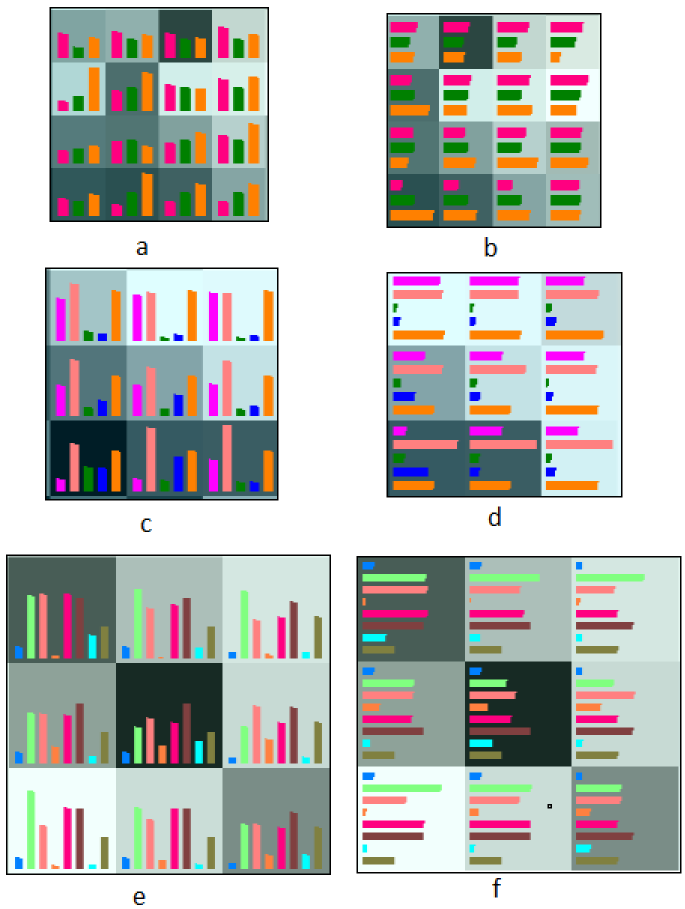

Figure 5.12: Three icon sizes for different numbers of icon elements. a-b: Icon size of 35 by 35 pixels for three or less attributes. c-d: Icon size of 50 by 50 pixels for four to six attributes. e-f: Icon size of 70 by 70 pixels for seven to nine attributes.

Both Figure 5.12a and b show the icon configuration of 35 by 35 pixels. Each icon element was sized taking into account the icon size, number of icon elements, and the attribute value. It is hard to state objectively whether the horizontal or the vertical bar configuration is better, rather the decision is based on user preference. In Figure

5.12c and d, the icon size is 50 by 50 pixels and each icon displayed five attributes simultaneously. The last two images in Figure 5.12 show the largest icon at 70 by 70 pixels, which contains up to nine attributes. Compared to the IconMapper System, the new icon design is more flexible and allows for more attributes to be visualized. In addition, the GeoIcon Viewer efficiently utilized all the available icon pixel space as shown in Figure 5.12. A large section of an icon was often empty in the IconMapper System where the position and size of icon elements were fixed.

Users generate an icon image map by selecting a location on the overview map once the attributes and their colours have been selected. The following subsections show multiple icon image maps with different icon sizes, as well as the corresponding image layers charts. Two sets of input data were used for each icon configuration. The first dataset, shown in Figure 5.2 to 5.9, has a binary data range. The binary images were created due to the high threshold value required in order to filter out the data noise. A second input dataset with a greater data range was required to demonstrate how the size of an icon element changed with different attribute values. The binary input dataset is referred to as $Input_2$ and the dataset with a range of fifteen is referred to as $Input_{15}$ to try and make the evaluation process easier to follow.

5.2.4 Visualization Outputs: Three Attributes

Figure 5.13 displays the icon image map for dataset $Input_2$. The three minerals chosen as input were alunite, hydroxide, and kaolinite. The focus region, which is in the center of the copper porphyry, is marked by the target box in the overview map. The only possible icon element sizes are either the full bar or none due to the binary data range. Half of the icons in Figure 5.13 contain no icon elements, which based on mineral maps shown in the previous section, seems to be a correct depiction of the distribution pattern at this location. There are roughly forty-five icons that contain

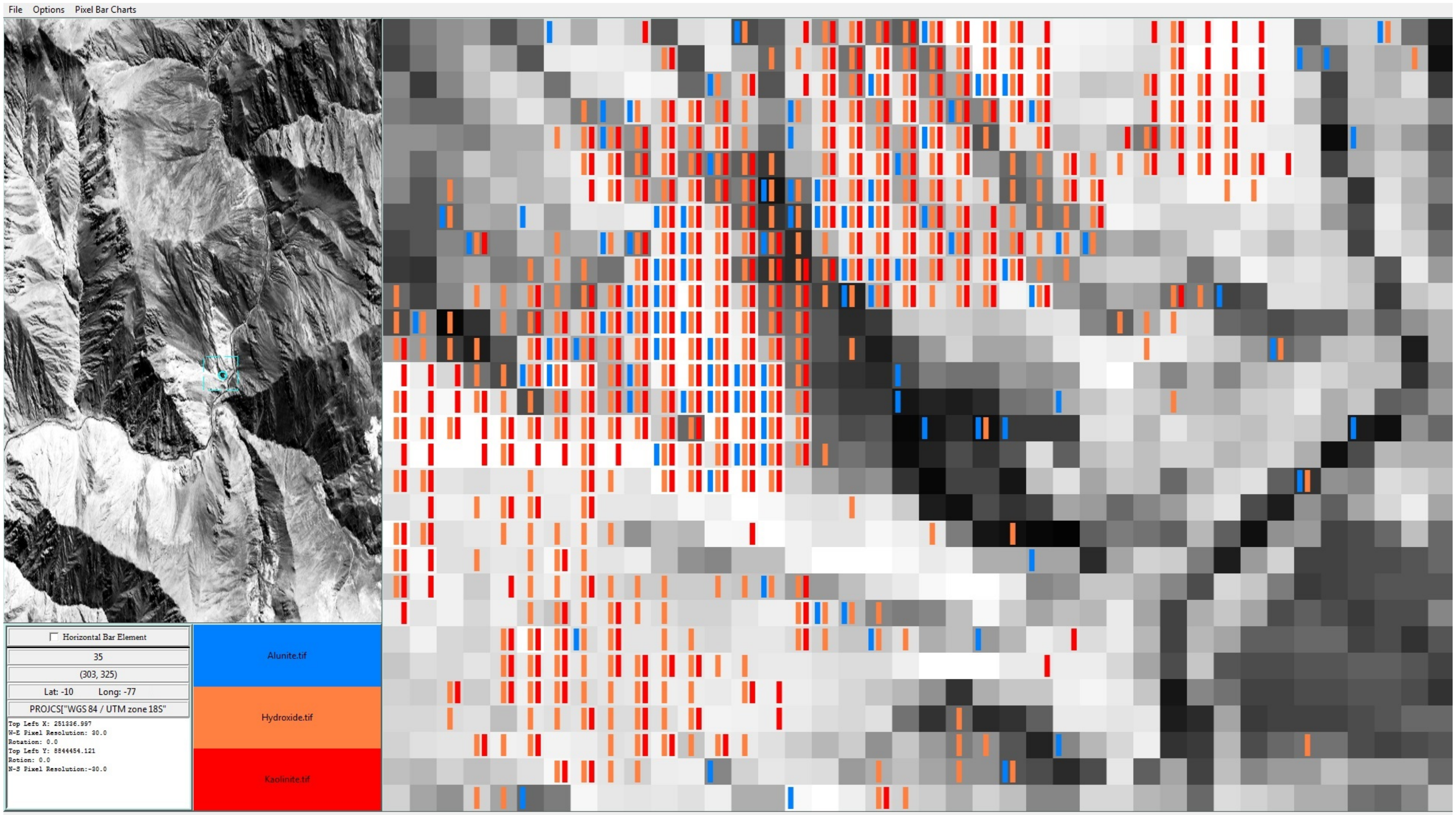

Figure 5.13: An icon image map where each icon displays three attributes. The background colour of each icon may convey additional information about a particular location depending on the overview image used.

all three minerals. These are potential location of interests that might warrant further in-depth analysis by the user. Furthermore, locations with all three minerals seem to have relatively higher reflectance values as denoted by a lighter icon background colour. Higher reflectance values at these locations are most likely associated with clay alterations and leached surfaces.

As shown in Figure 5.2 and 5.5, there appears to be alunite and hydroxide along river banks, which potentially indicates the movement of minerals via weathering, erosion, and deposition. There are few icons with blue or orange icon elements on the right side of the icon image map shown in Figure 5.13 which corresponds with what is depicted in Figure 5.2 and 5.5. The distribution patterns of the three minerals are also captured by the icon image map. The low amount of alunite within the copper porphyry is reflected by the small number of icons with a blue icon element. In contrast, over half of all icons contain hydroxide and/or kaolinite which corresponds to what the mineral index images have displayed.

Figure 5.13 shows the icon image map for $Input_2$ and only two icon element sizes were rendered due to the data range. Figure 5.14 shows the icon image map output with the $Input_{15}$ dataset. The location of both icon image maps is the same, but since the second input dataset has a data range of fifteen values, there are fifteen possible icon element sizes rendered within the icon image map. Looking at Figure 5.14, there are more non-empty icons but the patterns illustrated in Figure 5.13 are still present. The ratio between the amount of alunite compared to hydroxide or kaolinite can still be seen in Figure 5.14. Furthermore, the apparent movement of alunite and hydroxide from the ridges and slopes to the river banks is also visible on the icon image map.

The range of the $Input_{15}$ dataset was altered from 241-255 to 0-14 which increased how much the icon element size changed per unit value. If the data range was unaltered, the size difference between icon elements with value 241 and 242 is nearly indistinguishable as their standardized values, calculated via Equation (4.3) is equal to 0.945098 and 0.949019 respectively. For demonstration purposes, the data range was altered to allow icon sizes to be more visually distinguishable. However as the data range or the maximum value increases, the icon elements become harder to compare visually. Based on Tobler's First Law of Geography (Tobler, 1970) the value range of the input data within a neighbourhood is comparatively small and thus results in icon elements that are similar in size. The Region-of-Interest Image Layers Chart method is designed to complement the GeoIcon Image Map method when icon element size differences are indistinguishable. The statistics of the input matrix is used to re-standardize the attribute value of each raster cell via Equation (4.3). Rather than using size to represent quantity, a colour ramp is applied to each attribute image layer bar, where the new maximum standardized value is assigned the icon element colour and the minimum value is coloured white. The result is shown in Figure 5.15.

The three image layer bars in Figure 5.15 represent the same three attributes at the same location as in Figure 5.13 and 5.14. The Region-of-Interest Image Layers Chart output (image layers chart) illustrates the same pattern as the icon and index image maps. Unlike the icon image map, it is much easier to see how the mineral levels fluctuate. For hydroxide and kaolinite, the main clusters do not have a high value variation as indicated by the solid orange and red in the center. Moving away from the center of the clusters, the kaolinite and hydroxide levels start to decrease. The distribution pattern of alunite is similar where there is a smaller cluster near the center and attribute values decrease with distance away from the center.

Explanation about distribution patterns of kaolinite, alunite, and hydroxide are not the purpose of the GeoIcon Viewer which is a visual exploratory environment that only seeks to detect, uncover, and depict, but not explain, patterns. Furthermore, the goal of the case study was to use real world data to demonstrate and evaluate the

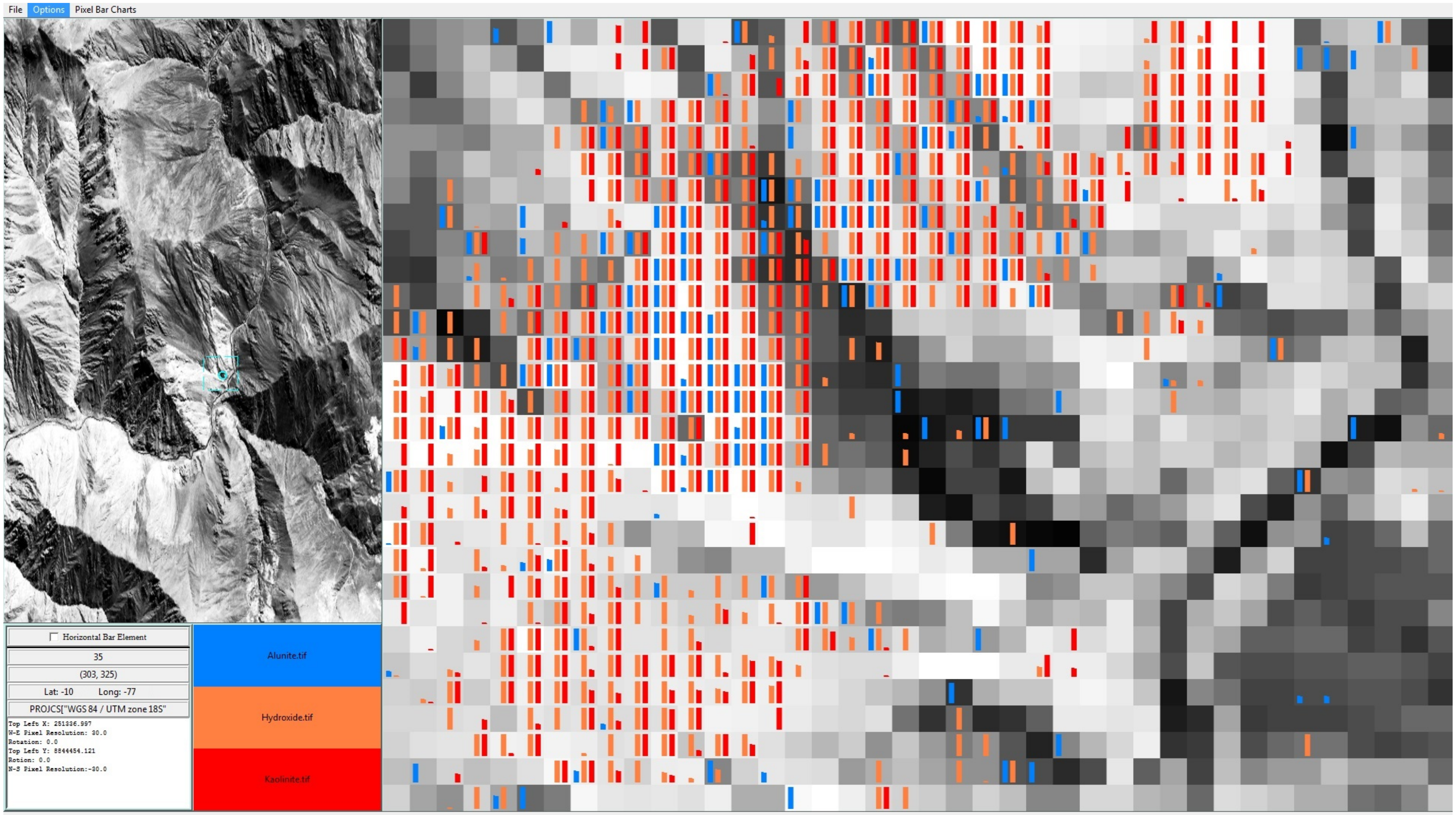

Figure 5.14: The icon image map of the second dataset, $Input_{15}$, where a lower threshold value results in more icon element size configurations.

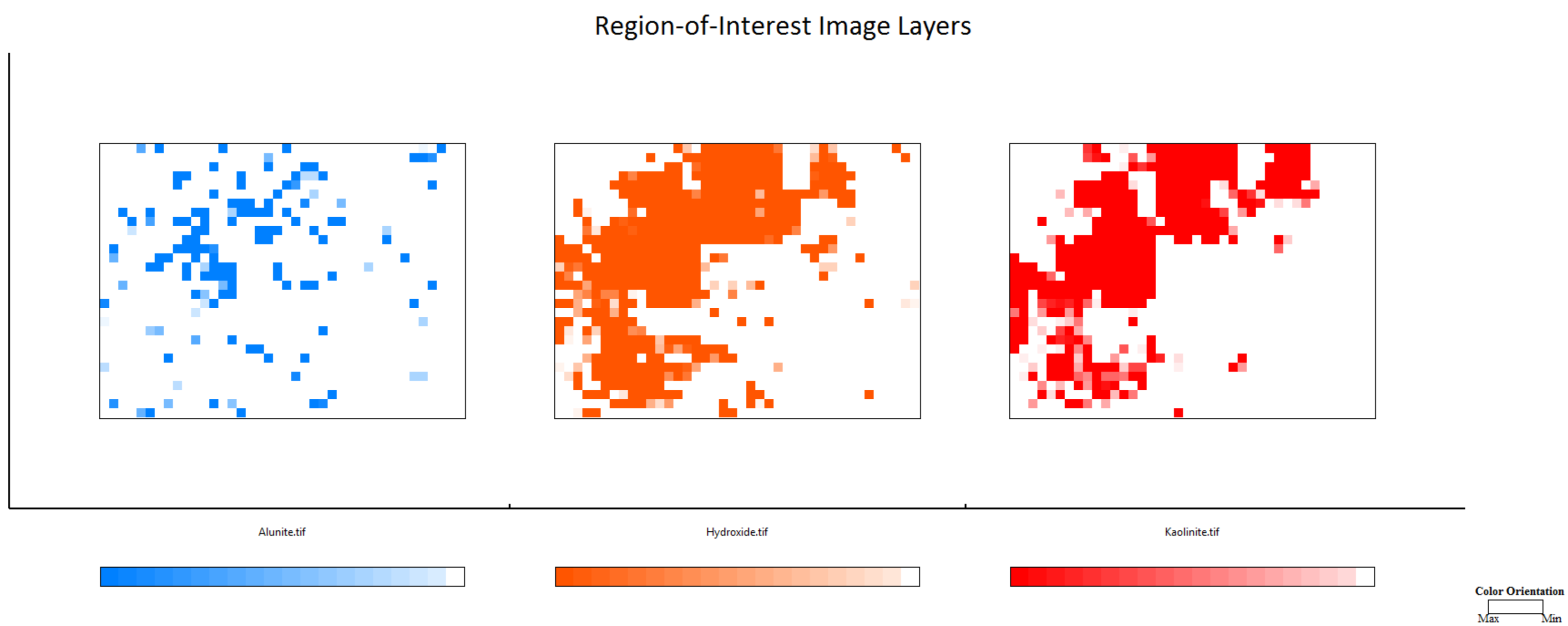

Figure 5.15: Region-of-Interest Image Layers Chart output for alunite, hydroxide, and kaolinite. Colour is used to represent quantity rather than icon element size.

software. The icon configuration for visualizing up to three attributes simultaneously was presented in this subsection. It can be stated that this icon configuration was successfully implemented based on Figure 5.13 and 5.14. Furthermore, the Region-of-Interest Image Layers Chart method for three attributes was also successfully implemented. The next section presents the GeoIcon Image Map and Region-of-Interest Image Layers Chart visualization outputs for five attributes.

5.2.5 Visualization Outputs: Five Attributes

The previous section displayed the visualization outputs for three attributes. By adding more attributes, both the number and complexity of potential patterns increase. In this section, two additional attributes were selected as input data: carbonate and quartz. The resulting icon image maps are shown in Figure 5.16 and 5.17. The size of icons increased from 35 by 35 pixels to 50 by 50 pixels with a trade-off in the number of raster cells visualized.

The icon image map presented in Figure 5.16 used the input dataset $Input_2$. The upper left region of the new icon image map is the location of the main cluster of kaolinite and hydroxide shown in Figure 5.13 and 5.14. The central cluster of kaolinite and hydroxide does not seem to have much overlap with quartz or carbonate. The overlap between different minerals occurs at the edge of the kaolinite and hydroxide cluster. The predominate minerals present in this region are still hydroxide and kaolinite based on the icon image map. The apparent movement pattern of alunite and hydroxide from the ridges to river banks was not displayed as the coverage area of the icon image map was reduced due to number of attributes rendered and the icon size.

The mineral and icon image maps reveal that carbonate, quartz, and mafic have a

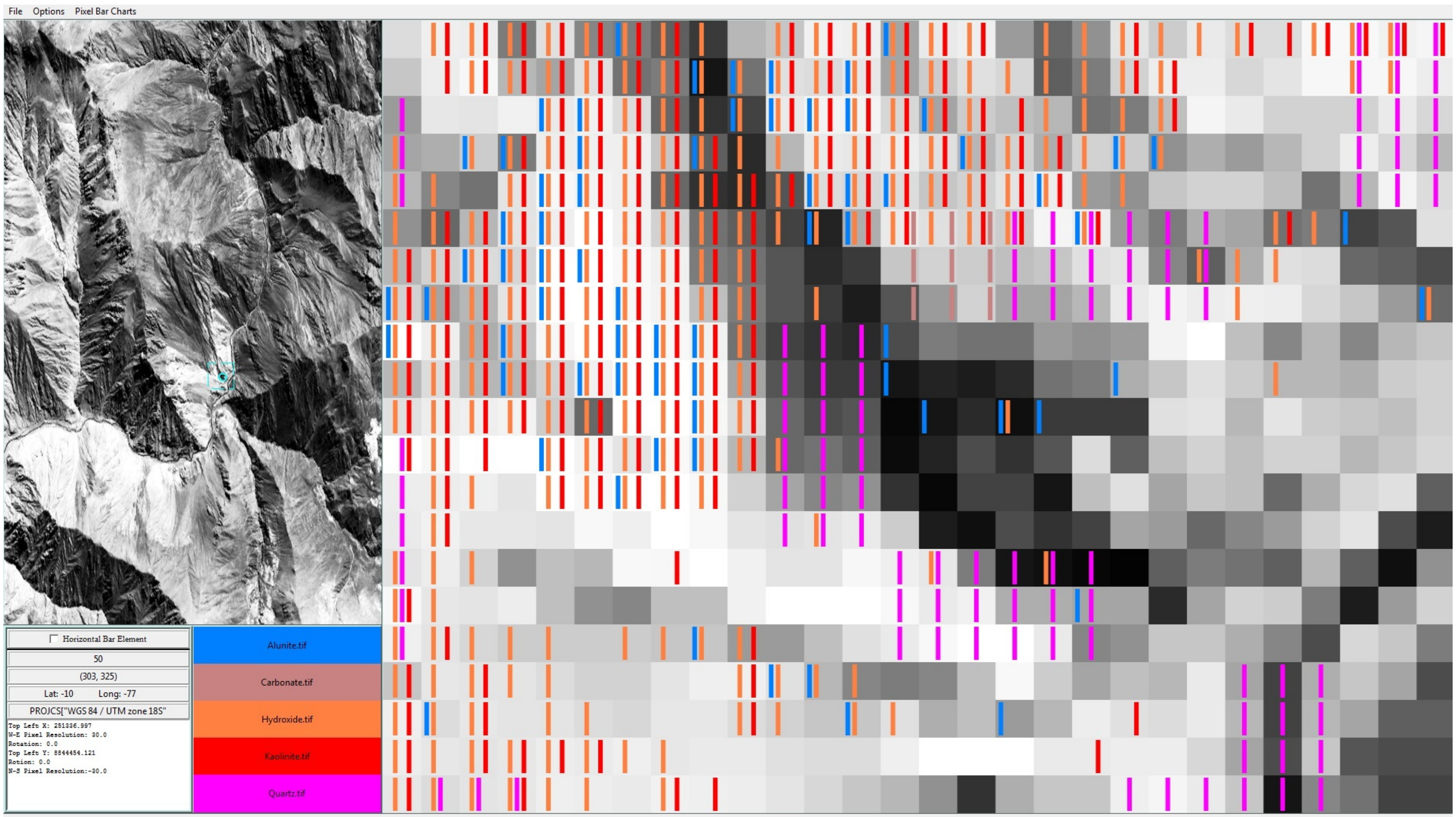

Figure 5.16: Icon image map for five attributes at same location as Figure 5.13 with a data range of 2.

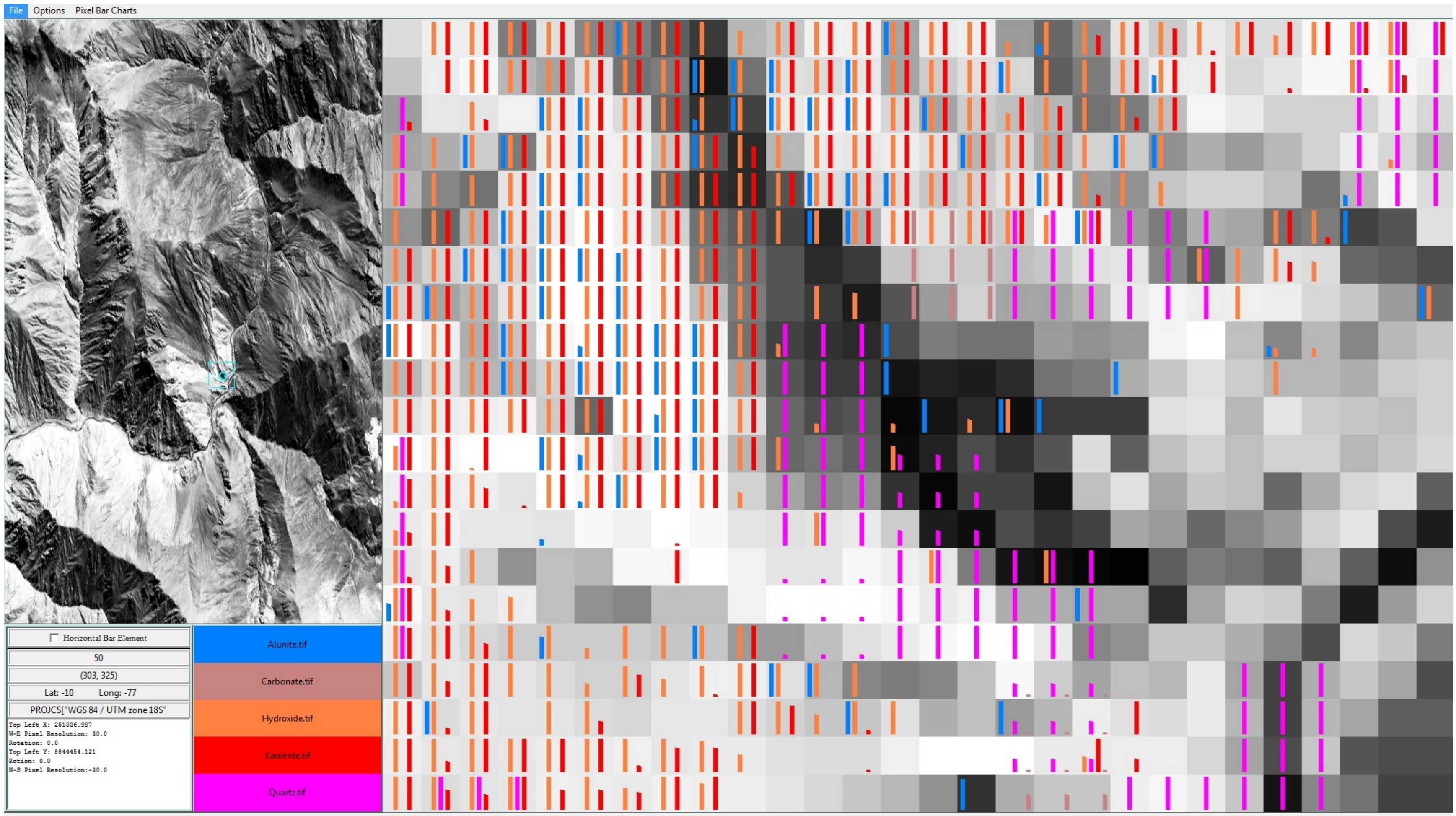

Figure 5.17: Icon image map of five attributes at same location as Figure 5.14 with a data range of 15.

blocky spatial distribution pattern. In the center of Figure 5.16, each quartz location is represented by a cluster of 18 adjacent icons in a 3 by 6 or a 6 by 3 configuration. Carbonate was represented by a cluster of 3 by 3 adjacent icons. The block effect was caused by the thermal infrared bands that were used to create the three mineral layers (Table 5.1). The bands had a coarse resolution of 90 meters. In order to create index layers with the same spatial resolution and extent, the thermal infrared bands were resized or magnified by a factor of three. Each pixel in the original band image was duplicated by a set amount in both X and Y directions to create a 30 meters resolution image. Thus, each pixel in the original image corresponds to a 3 by 3 matrix on the output image which is then filled with the value of the original pixel. This process led to the block effect seen Figure 5.16.

Figure 5.17 shows the icon image map output with the $Input_{15}$ dataset. Patterns displayed in 5.17 are the same as the ones shown in Figure 5.16. The only difference is that there are more icons with different sized icon elements. The block effect in carbonate and quartz is also present in Figure 5.17. Based on the four icon image maps, a potential correlation exists between hydroxide and kaolinite, as most icons with hydroxide contain kaolinite. Due to the distribution patterns of the five minerals, only certain areas of the copper porphyry contained more than three minerals. The icons successfully visualized the multiple attributes for those areas.

Figure 5.18 shows the Region-of-Interest Image Layers Chart output (image layers chart) for five attributes. The block effect can be clearly seen in the carbonate and quartz image layers. Based on the colours in the two image layers, the value range for each mineral in this region is very small. Of the five attributes, only kaolinite and hydroxide seem to have a matching colour pattern.

This subsection showed the GeoIcon Image Map and Region-of-Interest Image Layers Chart with five input attributes. Only a few icons contained more than three

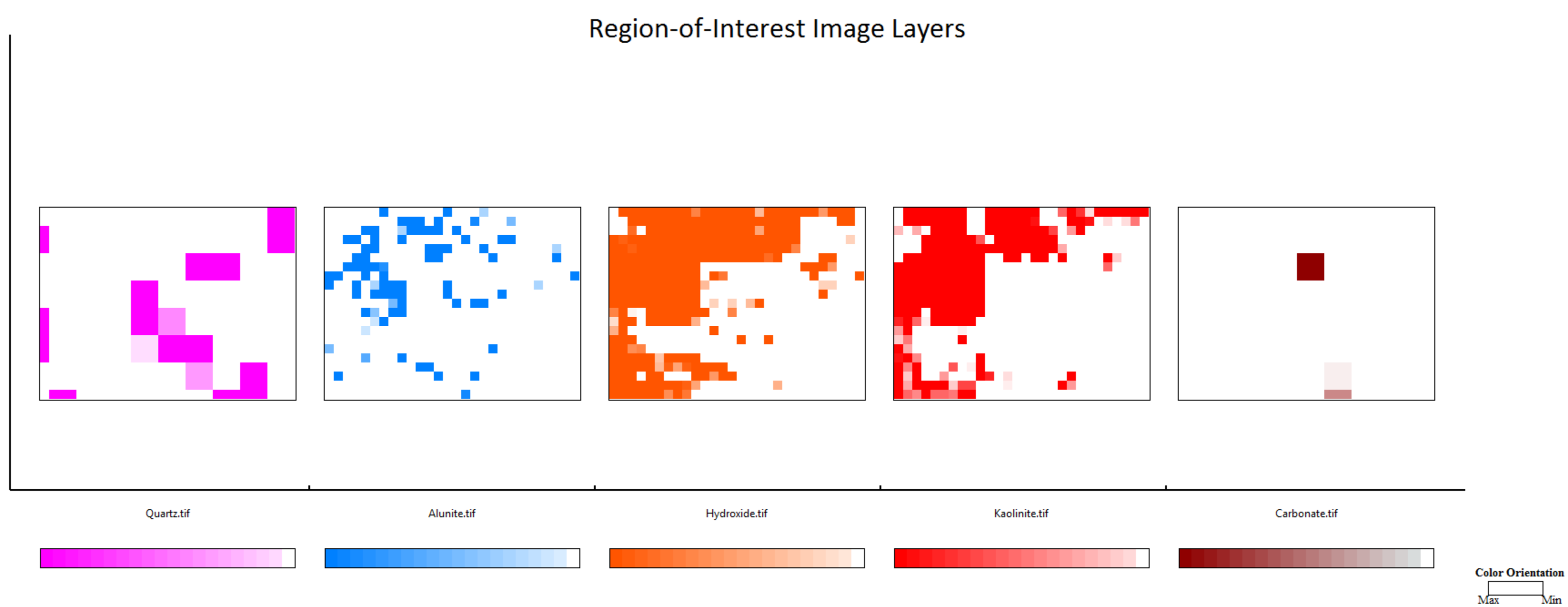

Figure 5.18: Image layers chart for quartz, alunite, hydroxide, kaolinite, and carbonate.

attributes due to the threshold value and the resulting mineral distributions. Based on Figure 5.16 and 5.17, the second icon configuration, which is for four to six attributes, was successfully implemented. Furthermore, Figure 5.18 demonstrated that the Region-of-Interest Image Layers Chart method was able to successfully visualize five attributes. The next subsection shows the visualization output for all eight attributes. A third un-thresholded input dataset was used due to the sensitivity of the input data to the threshold value.

5.2.6 Visualization Outputs: Eight Attributes

The aim of this section is to demonstrate the capability of the GeoIcon Viewer in visualizing eight attribute simultaneously. The input dataset was not thresholded and thus contained a lot of noise. The data noise is not too critical of an issue since the purpose of this particular section was to demonstrate and evaluate the GeoIcon Viewer.

The GeoIcon Viewer was designed to visualize up to nine attributes simultaneously. The icon size changes to 70 by 70 pixels for seven to nine attributes, and the total number of raster cells visualized in the icon image map and image layers chart was reduced. Figure 5.19 and 5.20 show the resulting icon image map and image layers chart respectively.

The input data used in Figure 5.19 was not thresholded and every icon contains eight icon elements. It is pointless to try and interpret this particular icon image map as most icon elements represent noise. However, Figure 5.19 does demonstrate that the GeoIcon Viewer is able to successfully visualize eight attributes. The new query feature, as seen in Figure 5.19, returned the attribute name, standardized value, and the raw data value via mouse click on an icon element. The implementation of the new query feature appears as a small text box near the left edge of the icon image map shown in Figure 5.19.

Figure 5.20 shows the corresponding Region-of-Interest Image Layers Chart output for eight attributes. There is a secondary query system implemented for the image layers chart that returned the input standardized value. Even though the data used do not allow for pattern detection due to the noise, the un-thresholded data was used to demonstrate the successful implementation of the two visualization methods in the GeoIcon Viewer for eight attributes.

While Figure 5.19 and 5.20 demonstrated the successful implementation of the GeoIcon Image Map and the Region-of-Interest Image Layers Chart, they also highlighted the hindrance in visual analysis with too many attributes visualized simultaneously. An intuitive attribute colour representation, accompanying data query feature, and a legend help to make the interpretation process more streamlined and feasible. However, there is still a limit on how many attributes can be effectively visualized. Just as in Figure 2.7, too much information displayed overwhelms users and hinders the exploratory data analysis process. The correct number of attributes to visualize is subjective and varies based on the user, data, application, objective, and the graphic design. However, the GeoIcon Viewer allows users the option to choose how many attributes to visualize.

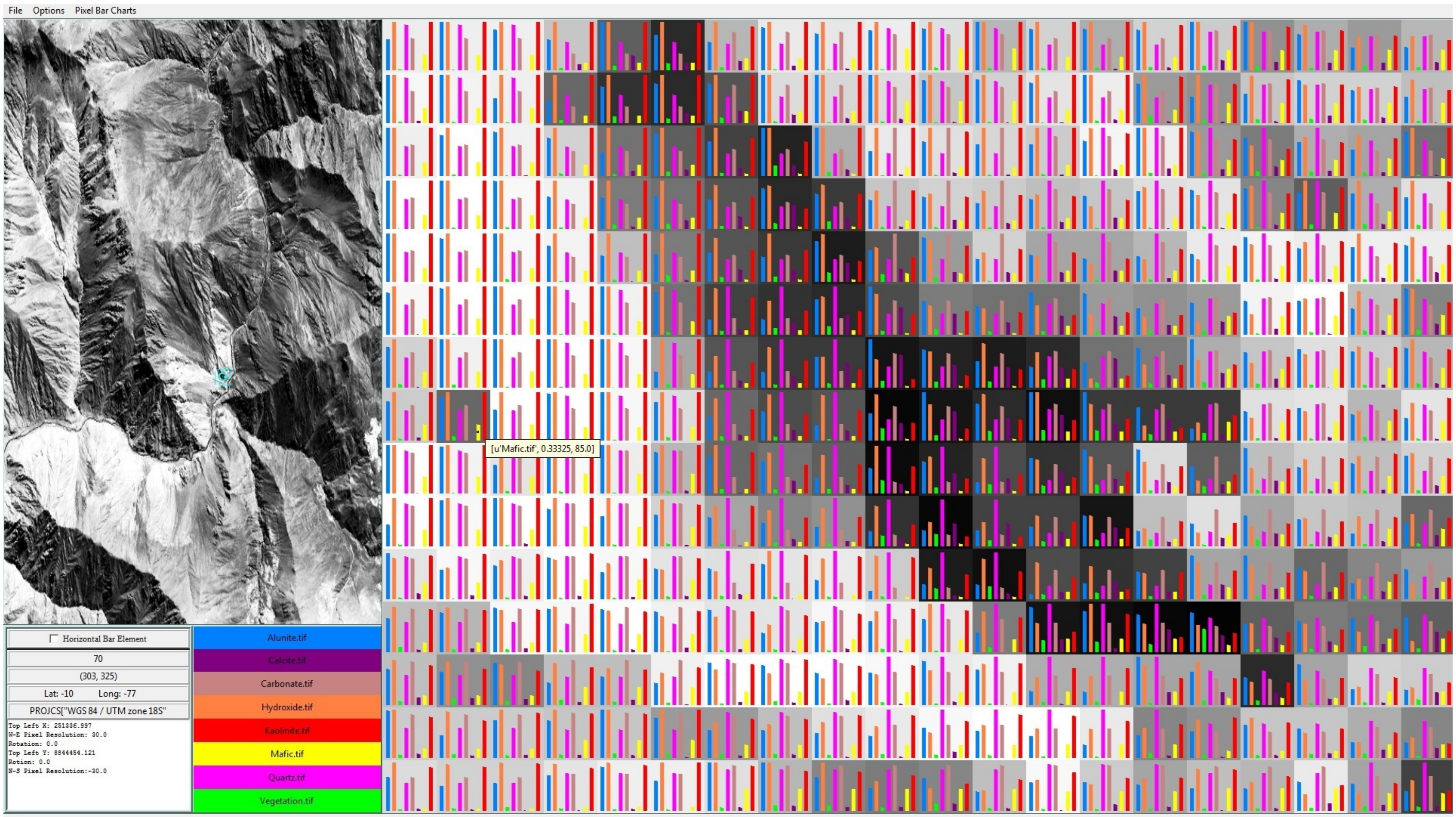

Figure 5.19: Icon image map for eight attributes. The input data was not thresholded and every icon has 8 icon elements.

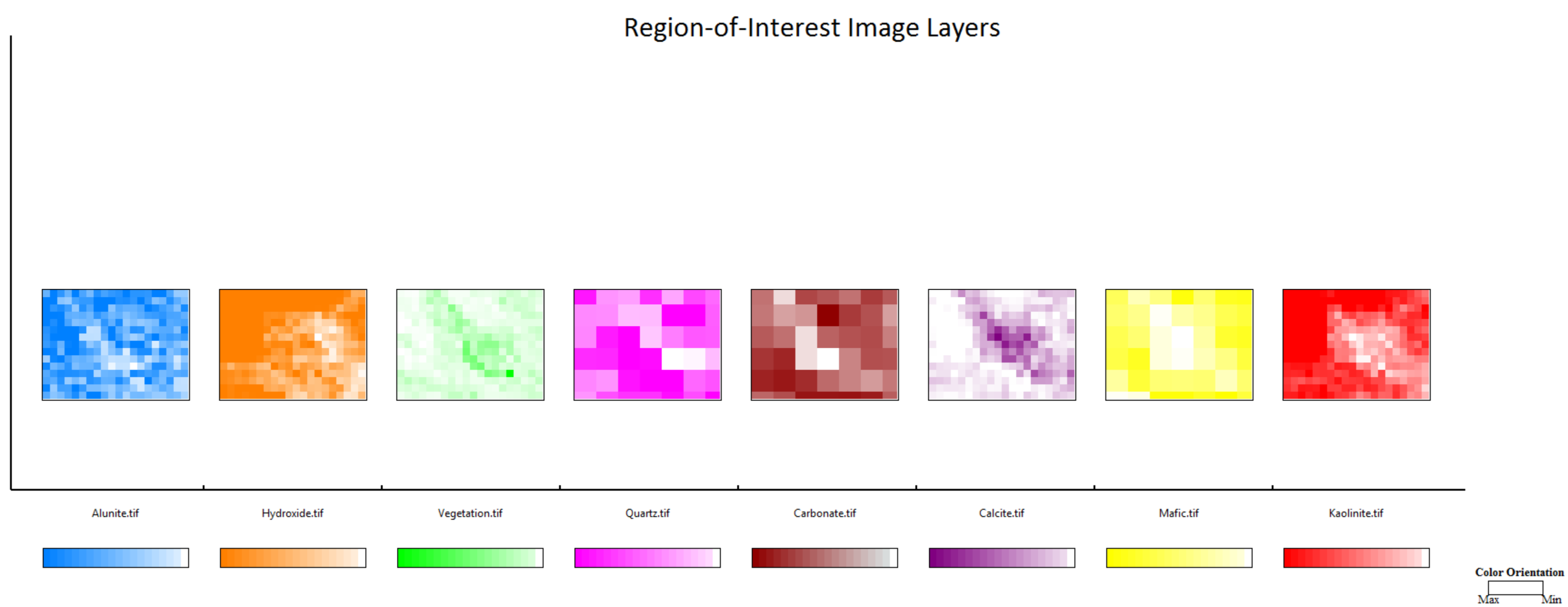

Figure 5.20: Region-of-Interest Image Layers Chart output for all eight attributes.

5.3 Program Evaluation

There are two types of evaluation that can be performed for a software program. The first type of evaluation examines the GeoIcon Viewer's desired functions and its actual capabilities. The second evaluation addresses the effectiveness of the GeoIcon Viewer and its two visualization methods. Based on the System Design chapter, the program must have a new icon design with automatic adjustments in size to reduce unwanted visual effects and to utilize the available pixel space effectively. As demonstrated in this chapter, the program dynamically resizes the icons and icon elements based on the number of attributes being displayed. Furthermore, the number of attributes that can be visualized increased from five to nine. Lastly, users now have the ability to set the colour of each icon element for a more intuitive colour representation. Thus, the new icon design and the GeoIcon Image Map method are successfully implemented. The GeoIcon Viewer created and displayed an explorable and interactive icon image map 'on the fly' or in real time rather printing out a static image file shown in Figure 5.14 or 5.17.

Another required feature was the Region-of-Interest Image Layers Chart method. This method was designed to complement the GeoIcon Image Map visualization approach by highlighting small attribute value differences among raster cells through the use of colour. The Region-of-Interest Image Layers Chart method was implemented successfully and the image layers chart was useful in pattern detection and confirmation as shown in Figure 5.15 and 5.18.

The dynamic query system was designed to replace the query function in the IconMapper System. The new query feature, as shown in Figure 5.19 returned the attribute name, standardized value, and raw data value on demand via mouse click. A secondary query feature was also implemented for the Region-of-Interest Image Layers

Chart method, and returned the standardized value of each input raster cell with a mouse click. The goal of the query system, along with the legend and colour picker, was to support the visual interpretation process.

The three program features mentioned above were all successfully implemented as demonstrated in the section above. There were also many minor required design features listed during the design stage such as TIFF file processing and program portability. The program was able to successfully read, access, and retrieve TIFF files to support critical program features. For instance, image to geographic coordinate conversion and creating image subsets. Whereas the older IconMapper System required Window OS and a third party software to execute, the GeoIcon Viewer was able to run as a stand-alone program on Window 7, Windows 8, and Mac OS X.

The second type of evaluation concerns the effectiveness of the visualization software. It is hard to evaluate a subjective topic such as effectiveness. How does one measure the effectiveness of a visualization method in stimulating the human cognition system? Without comprehensive surveys, the effectiveness of any particular visualization method or program is hard to quantify without considerable bias. The scope and time frame of the project did not allow for the survey option. Evaluating the intuitiveness of a program is difficult as well. To make an intuitive and easy to learn program, the GeoIcon Viewer does not have too many options, nor complicated GIS methods. To the experienced GIS analysts, the program may be lacking. However, it might be exactly what novice users require. Thus it is hard to evaluate and compare the effectiveness and intuitiveness of different visualization methods which were designed with different purposes in mind. Choosing a particular visualization method requires a clear understanding of the project including the objectives and input data. The next chapter contains a summary, concluding remarks, and potential future improvements.

Chapter 6

Summary and Future Work

Data mining or Knowledge Discovery from Database is a multi-stage process designed to uncover hidden patterns from databases that are vast and complex using a mixture of computational and visual methods. Computational methods offer an automated and efficient search for numerical patterns whereas visual methods aim to engage the human cognition ability to recognize patterns. Visual approaches do not place an *a priori* assumption on potential patterns that might exist within a database and thus reduce the likelihood of a pattern being discounted and remaining undiscovered.

Current geovisualization methods are mainly designed for uni- and bivariate spatial data, but the increasing complexity of spatial databases requires the development of multivariate spatial data visualization. Pazner and Zhang (2004) used an icon approach where each raster cell is replaced by an icon that renders and visualizes multiple attributes through visual components such as colour and size. However, there were a few issues with the IconMapper System that limited its effectiveness. The implemented icon design was only able to visualize five attributes, and had a static icon element positioning system which led to unwanted visual effects. In addition, the lack of icon element colour customization and an effective data query system created an unintuitive visual representation.

The goal of this project was to develop new visualization software based on the icon visualization method used in IconMapper System. The new software, named GeoIcon Viewer, improves on the IconMapper System by incorporating an improved icon design (GeoIcon Image Map), dynamic data query system, and the Region-of-Interest Image Layers Chart method.

The design process of the GeoIcon Viewer was based on considerations of input data, visualization methods, and interaction techniques. A set of required components including GUI objects, data objects, and functions were specified in a manner that addresses these three considerations. Python was used to develop and implement each individual component following the Object Oriented Programming paradigm. Each component is a class containing different methods that interact with other classes. Components were integrated to form the GeoIcon Viewer after each component was implemented and tested.

A case study was used to demonstrate and evaluate the GeoIcon Viewer. Mineral and vegetation index images were created from ASTER remote sensing data using band ratio and thresholding operations. Each major program component was evaluated and compared to the design requirements. For the input data requirement, the program was able to process TIFF images and supported the display of both grayscale and colour images. Furthermore, the program supported the creation and display of false colour composite images. The GeoIcon Viewer utilized the spatial projection and georeference coefficients to provide the spatial location of user queries. The visualization methods used in the GeoIcon Viewer are the GeoIcon Image Map and the Region-of-Interest Image Layers Chart methods. Both methods were successfully implemented and helped a user to visually uncover potential patterns of multiple mineral abundances. From the user interaction aspect, the software allowed users to query location, attribute name, raw input value, and standardized value from the icon image map. The brushing and linking technique discussed in Chapter 3 was also implemented successfully. Changes to the attribute selection, icon element colour, and region of interest were reflected by the visualization output in the Visualization Window. Lastly, the GeoIcon Viewer is more portable than the IconMapper System since the GeoIcon Viewer does not rely on third party software and is able to run under the Window, Unix, and Linux operating systems. Based on evaluation results shown in Chapter 5, all of the desired features including the new icon design, query system, and the Region-of-Interest Image Layers Chart method were successfully implemented and are fully functional.

The thesis project answered the research question *How can spatial multivariate data be visualized using an icon based non-fused co-visualization approach?* The GeoIcon Viewer is a portable interactive visualization program that uses an icon based non-fused co-visualization approach, and allows users to explore interrelationships between up to nine spatial variables location by location.

While the GeoIcon Viewer and its two visualization methods were successfully implemented, the icon image map, in particular, illustrated the weakness of the icon and current geovisualization methods. The number of attributes visualized has a big impact on the effectiveness of a visualization method. The information density of a visualization output grows exponentially with the number of attributes. Visual attributes such as colour, size, texture, and pattern may be insufficient to handle the ever growing dimensionality of spatial databases and the complexity of geographic patterns. Other human senses like hearing can be incorporated into the non-computational mining process.

6.1 Future Improvements

Although all of the design features listed in the System Design chapter were successfully implemented, the GeoIcon Viewer is by no means finished. Knowing how effective or useful the program is for actual users is critical for future development. Without feedback from users about the program, it is difficult for the developer to determine if the visualization method or software application environment is effective in uncovering hidden patterns from very large databases.

Besides testing for the effectiveness of the program, there are many other features that can be added to create a better visual exploratory environment. From the input data aspect, the program lacks the capability to handle JPEG image files. More specifically, the program cannot process .JGW file which contains the spatial information for a JPEG image file. The program is also incapable of handling and displaying vector data. Having roads, rivers, or field polygons on display would provide additional spatial or locational context for users. Another area that needs improvement is the zoom feature. A zoom function can be attached to the Overview Window which allows users to zoom first on the overview image and then select regions of interest. Another potential program feature is the ability to render GeoIcons and Image Charts at meso and macro levels that cover larger areas. This feature would take advantage of high resolution monitors.

Changes to already implemented features such as visualization methods or the data query feature may also improve the effectiveness of the program. For the GeoIcon Image Map method, more options can be designed and implemented including icon element colour ramp, icon element ordering by size, and even different geometric shapes and orientations for icon elements. The option to remove the icon background layer at the neighbourhood scale display may help to reduce the information density of the icon map. The Region-of-Interest Image Layers Chart visualization method may be improved by adding a ranking feature for image layer bars. The height of the each image layer bar can be set based on spatial or non-spatial statistics of the input matrix, and this feature allows the image layers chart to convey more information to users in a simple but effective way. The query feature can be also improved upon by changing its user interface and increasing the query scope. The standardization formula in Equation (4.3) is the maximum score method for maximization criterion. Providing the option to use: (1) cost minimization equation, (2) score range method, or (3) no standardization, would create a more flexible program. Lastly, changes to user interface elements such as legend layout, text size, and window spacing will improve the user friendliness of the program.

With feedback from users and the implementation of additional program features, future versions of the GeoIcon Viewer could be more capable and provide better visual exploratory analysis in the context of the Geographical Knowledge Discovery from Databases process.

Bibliography

Anscombe, F. J. (1973). Graph in statisitical analysis. *The American Statistician*, 27(1):17–21.

Chernoff, H. (1973). Using faces to represent points in k-dimensional space. *Journal of the American Staistical Association*, pages 361–368.

Dykes, J. A. (1997). Exploring spatial data representation with dynamic graphics. *Computer and Geosciences*, 23(4):345–370.

Fairbairn, D., Andrienko, G., Andrienko, N., Buziek, G., and Dykes, J. (2001). Representation and its relationship with cartographic visualization. *Cartography and Geographic Information Science*, 28(1):13–28.

Fayyad, U., Shapiro, G. P., and Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of ACM*, pages 27–34.

Fekete, J.-D., van Wijk, J. J., Stasko, J. T., and North, C. (2008). The value of information visualization. In Kerren, A., Stasko, J. T., Fekete, J.-D., and North, C., editors, *Information Visualization: Human-Centered Issues and Perspectives*, pages 1–18. Springer.

Fotheringham, A. S. and Charlton, M. (1994). GIS and exploratory spatial data analysis: An overview of some research issues. *Geographical Systems*, 1:315–327.

Gahegan, M. (1998). Scatterplots and scenes: Visualization techniques for exploratory spatial analysis. *Computer, Environment and Urban Systems*, 22(1):43–56.

Gahegan, M., Wachowicz, M., Harrower, M., and Rhyne, T. (2001). The integration of geographic visualization with knowledge discovery in databases and geocomputation. *Cartography and Geographic Information Science*, 28(1):29–44.

Guo, D. (2003). Coordinating computational and visualization approaches for interactive feature selection and multivariate clustering. *Information Visualization*, pages 232–246.

Guo, D., Chen, J., MacEachren, A. M., and Liao, K. (2006). A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE Transactions on Visualization and Computer Graphics*, 12(6).

Guo, D., M. MacEachren, A. M., and Zhou, B. (2005). Multivariate analysis and geovisualization with an integrated geographic knowledge discovery approach. *Cartographic Geographic Information Science*, pages 113–132.

Hearnshaw, H. (1994). *Psychology and Displays in GIS*, chapter 21, pages 193–211. Wiley.

Hernandez, T. (2007). Enhancing retail location decision support: The development and application of geovisualization. *Journal of Retailing and Consumer Services*, 14:249–258.

Hinneburg, A., Keim, D. A., and Wawryniuk, M. (1999). HD-Eye: Visual mining of high dimensional data. *Computer Graphics and Applications, IEEE*, 19(5).

Imhof, E. (1982). *Cartographic Relief Presentation*. Walter de Gruyter, Berlin, Germany.

Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*, pages 69–97.

Keim, D. A., Hao, M. C., and Dayal, U. (2002). Hierarchical pixel bar charts. *IEEE Transactions on Visualization and Computer Graphics*, 8(3).

Keim, D. A. and Hermann, A. (1998). The gridfit algorithm: An efficient and effective approach to visualize large amounts of spatial data. *IEEE Visualization '98 Proceedings*, pages 181–188.

Keim, D. A. and Kriegel, H.-P. (1996). Visualization techniques for mining large databases: A comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):923–938.

Keim, D. A., Panse, C., and Sips, M. (2005). Information visualization: Scope, techniques and opportunities for geovisualization. In Dykes, J., Maceachren, A. M., and Kraak, M.-J., editors, *Exploring Geovisualization*, pages 23 –53. Elsevier.

Kim, S., Maciejewski, R., Malik, A., Jang, Y., Ebert, D. S., and Isenberg, T. (2013). Bristle maps: A multivariate abstraction technique for geovisualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1438–1454.

Kimerling, A. J., Buckley, A. R., Muehrcke, P. C., and Muehrcke, J. O. (2012). *Map Use Reading Analysis Interpretation*. ESRI Press Academic, Redlands, California.

Lo, C. and Yeung, A. K. (2007). *Concepts and Techniques of Geographic Information Systems*. Pearson Prentice Hall, 2nd edition.

Lutz, M. (2013). *Learning Python*. O'Reilly, 5th edition.

MacDougall, E. B. (1992). Exploratory analysis, dynamic statistical visualization, and geographical information systems. *Cartography and Geographic Information Systems*, 19(4).

MacEachren, A. M. and Kraak, M.-J. (2001). Research challenges in geovisualization. *Cartography and Geographic Information Science*, 28(1):3–12.

Malczweski, J. (1999). *GIS and Multicriteria Decision Analysis*. John Wiley & Sons Inc.

Miller, G. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97.

Miller, H. J. and Han, J. (2009). Geographic data mining and knowledge discovery: An overview. In Miller, H. J. and Han, J., editors, *Geographic Data Mining and Knowledge Discovery*. CRC Press, 2nd edition.

Ninomiya, Y., Fu, B., and Cudahy, T. J. (2005). Detecting lithology with advanced spaceborn thermal emission and reflection radiometer (ASTER) multispectral thermal infrared 'radiance-at-sensor"data. *Remote Sensing of Environment*, 99:127–139.

Pazner, M. and Zhang, X. (2004). Icon image map technique for multivariate geospatial data visualization. *Cartography and Geographic Information Science*, pages 29–41.

Pour, A. B. and Hashim, M. (2012). The application of aster remote sensing data to porphyry copper and epithermal gold deposits. *Ore Geology Reviews*, 44:1–9.

Purchase, H. C., Andrienko, N., Jankun-Kelly, T., and Ward, M. (2008). In Kerren, A., Stasko, J. T., Fekete, J.-D., and North, C., editors, *Information Visualization: Human-Centered Issues and Perspectives*, pages 46–64. Springer.

Spence, I. and Garrison, R. F. (1993). A remarkable scatterplot. *The American Statistician*, 47(1):12–19.

Tobler, W. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46:234–240.

Tufte, E. R. (1990). *Envisioning Information*. Graphic Press, Cheshire, Connecticut.

Tufte, E. R. (1997). *Visual Explanations: Image and Quantities, Evidence, and Narrative*. Graphic Press, Cheshire, Connecticut.

Wise, S. (2002). *GIS Basics*. Taylor & Francis.

Worboys, M. and Duckham, M. (2004). *GIS: A Computing Perspective*. CRC Press, 2nd edition.

Curriculum Vitae

Name	Bo Shan
Place of Birth	P.R. China
Post-secondary Education	The University of Western Ontario London, Ontario, Canada M. Sc. September 2012 - December 2014
	The University of Western Ontario London, Ontario, Canada B.A September 2007 - May 2012
Related Work Experience	Teaching Assistant The University of Western Ontario September 2012 - May 2014
	GIS Technician A & L Canada Laboratories Inc. September 2012 - Present